

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

*Факультет інформатики та обчислювальної техніки  
Кафедра автоматизованих систем обробки інформації та управління*

"На правах рукопису"  
УДК 004.9

До захисту допущено  
В.о. завідувача кафедри  
Олександр ПАВЛОВ  
“        ”        20 20 р.

**МАГІСТЕРСЬКА ДИСЕРТАЦІЯ**

**на здобуття ступеня магістра**

**за освітньо-професійною програмою**

***«Інформаційні управляючі системи та технології»***

**зі спеціальності 126 *«Інформаційні системи та технології»***

**на тему:**

***«Інформаційна система нарахування бонусів співробітникам  
компанії»***

Виконала:

студентка VI курсу, групи ІС-92мп

Казмірчук Аліна Василівна

\_\_\_\_\_

Керівник:

доцент, к.т.н, доцент

Сперкач Майя Олегівна

\_\_\_\_\_

Консультант:

професор, д.т.н., доцент

Жаріков Едуард В'ячеславович

\_\_\_\_\_

Рецензент:

доцент, к.т.н, доцент

Писаренко Андрій Володимирович

\_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студентка \_\_\_\_\_

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

*Факультет інформатики та обчислювальної техніки  
Кафедра автоматизованих систем обробки інформації та управління*

Рівень вищої освіти – *другий (магістерський)*

Спеціальність – *126 «Інформаційні системи та технології»*

Освітньо-професійна програма *«Інформаційні управляючі системи та технології»*

В.о.завідувача кафедри

\_\_\_\_\_ Олександр ПАВЛОВ

«\_\_» \_\_\_\_\_ 2020 р.

**ЗАВДАННЯ  
на магістерську дисертацію студенту**

**Казмірчук Аліна Василівна**

1. Тема дисертації *«Інформаційна система нарахування бонусів співробітникам компанії»*, науковий керівник дисертації Сперкач Майя Олегівна, к.т.н., доцент, затверджені наказом по університету від «26» жовтня 2020 р. № 3132-с.

2. Строк подання студентом дисертації “ 02 ” 12 2020 р.  
\_\_\_\_\_

3. Об’єкт дослідження *Процес впровадження бонусної системи як складової програми мотивування співробітників*

4. Перелік завдань, які потрібно розробити *Виконати аналітичний огляд існуючих систем мотивування, систем планування та розподілу ресурсів, систем нарахування бонусів, методів складання планів нарахування; реалізувати алгоритми розв’язання задачі підтримки роботи системи нарахування бонусів; дослідити ефективність розробленого алгоритму; розробити програмну реалізацію системи нарахування бонусів;*

здійснити експериментальні дослідження розробленої інформаційної системи.

5. Орієнтовний перелік графічного (ілюстративного) матеріалу Схема структурна класів системи, схема структурна пакетів системи, схема структурна бази даних системи, схема структурна варіантів використання, графіки зміни множини рішень в процесі роботи алгоритму, екранні форми, блок-схема багатокритеріального генетичного алгоритму.

6. Орієнтовний перелік публікацій Три публікації: дві тези доповіді на наукових конференціях, одна стаття у фаховому виданні

7. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

8. Дата видачі завдання “ 01 ” вересня 20 20 р.

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Систематизація результатів огляду літератури	01.09	
2	Порівняльний аналіз існуючих методів розв’язання задачі	10.09	
3	Постановка та формалізація математичної моделі задачі	17.09	
4	Розробка алгоритму розв’язання задачі	20.09	
5	Розробка інформаційного та програмного забезпечення	11.10	
7	Проведення експериментальних досліджень розробленого алгоритму	12.11	
8	Оформлення документації	19.11	
9	Подання роботи на попередній захист	20.11	
10	Подання роботи на основний захист	02.12	

Студент

Аліна КАЗМІРЧУК

Науковий керівник

Майя СПЕРКАЧ

## РЕФЕРАТ

Магістерська дисертація: 97 с., 20 рис., 26 табл., 1 додаток, 60 джерел.

**Актуальність.** Успіх кожної компанії залежить від ефективності роботи її працівників. Розвитком теорії мотивації співробітників займаються вчені всього світу, зокрема Л.М. Гохберг, С.Д. Ілленкова, Є.А. Добролюбов. Найпоширенішою складовою системи мотивування є матеріальна. Відомо, що матеріальна винагорода виступає рушійною силою для мотивування працівників виконувати свої задачі якісно та прагнути до постійного покращення своїх навичок, підвищення професійної кваліфікації. Саме тому більшість компаній активно впроваджують бонусні системи для своїх працівників. Зокрема, найпоширенішою така практика є у відділах продажу та службах технічної підтримки. Однак дуже часто такі системи виявляються неефективними – виходить дуже дорого, але очікувані показники продуктивності не досягаються. Саме тому існує необхідність у створенні системи, яка б підтримувала процес правильного планування та розподілу бонусів, з метою, щоб бонусна система працювала ефективно та на благо компанії.

**Зв'язок роботи з науковими програмами.** Робота виконана на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Ефективні методи розв'язання задач теорії розкладів», державний реєстраційний номер 0117U000919.

**Мета дослідження** – підвищення ефективності роботи співробітників компанії з мінімальними витратами для компанії на бонусні виплати за рахунок впровадження інформаційної системи нарахування бонусів співробітникам компанії.

Для досягнення мети необхідно виконати наступні **завдання**:

- виконати аналітичний огляд існуючих систем мотивування, систем планування та розподілу ресурсів, систем нарахування бонусів, методів складання планів нарахування;
- реалізувати алгоритм розв’язання задачі підтримки роботи системи нарахування бонусів;
- дослідити ефективність розробленого алгоритму;
- розробити програмну реалізацію системи нарахування бонусів;
- здійснити експериментальні дослідження розробленої інформаційної системи.

**Об’єкт дослідження** – процес впровадження бонусної системи як складової програми мотивування співробітників.

**Предмет дослідження** – методи ефективного нарахування бонусів співробітникам для досягнення цілей компанії.

**Методи дослідження**, застосовані у даній роботі, базуються на методах теорії розкладів та багатокритеріальної оптимізації.

**Наукова новизна отриманих результатів** полягає в розробці математичної моделі для побудови ефективної бонусної системи мотивування співробітників компанії.

**Публікації.** Матеріали роботи опубліковані у XV міжнародній науково–практичній конференції «Математичне та імітаційне моделювання систем МОДС 2020» [1], на IV Всеукраїнській науково-практичній конференції молодих вчених та студентів «Інформаційні системи та технології управління» [2] та в науково-технічному журналі «Сучасні інформаційні системи» [3].

БАГАТОКРИТЕРІАЛЬНА ОПТИМІЗАЦІЯ, ЕВРИСТИКА, ГЕНЕТИЧНИЙ АЛГОРИТМ, ФРОНТ ПАРЕТО, ЗАДАЧА СКЛАДАННЯ РОЗКЛАДУ

## ABSTRACT

Master's dissertation: 97 p., 20 figures, 26 tables, 1 applications, 58 sources.

**Topicality.** The success of each company depends on the efficiency of its employees. The development of the theory of employee motivation is carried out by scientists around the world, in particular L.M. Gochberg, S.D. Illenkova, E.A. Dobrolyubov. The most common component of the motivation system is material.

It is known that material reward is the driving force to motivate employees to perform their tasks well and strive to constantly improve their skills, professional development. That is why most companies are actively implementing bonus systems for their employees. In particular, this practice is most common in sales departments and technical support services. However, very often such systems are inefficient - it is very expensive, but the expected performance is not achieved.

That is why there is a need to create a system that would support the process of proper planning and distribution of bonuses, in order for the bonus system to work effectively and for the benefit of the company.

**Relationship of work with scientific programs.** The work was carried out at the Department of Automated Systems for Information Processing and Management of the National Technical University of Ukraine "Kyiv Polytechnic Institute. Igor Sikorsky" within the framework of the theme "Effective methods for solving the problems of the theory of schedules ", state registration number 0117U000919.

**The purpose** of the study is to increase the efficiency of the company's employees with minimal costs for the company for bonus payments through the introduction of the Employee Benefits Accrual Information System.

**To achieve the goal**, you must accomplish the following tasks:

- perform an analytical review of existing motivation systems, planning and resource allocation systems, bonus accrual systems, accrual planning methods;
- implement algorithms for solving the problem of supporting the work of the bonus accrual system;
- to investigate the effectiveness of the developed algorithms;
- to develop a software implementation of the bonus accrual system;
- to carry out experimental researches of the developed information system;

**The object** of the research is the process of introduction of the bonus system as a component of the program of motivation of employees.

**The subject** of research is methods of effective accrual of bonuses to employees to achieve the company's goals.

**The research methods** used in this paper are based on the methods of schedule theory and multiobjective optimization.

**The scientific novelty** of the results obtained is the development of a mathematical model and algorithms for building an effective bonus system for motivating company employees.

**Publications.** Materials of the work were published in the XV International Scientific and Practical Conference "Mathematical and Simulation Modeling of MODS 2020 Systems", in the IV All-Ukrainian Scientific and Practical Conference of Young Scientists and Students "Information Systems and Management Technologies" and in scientific and technical Journal "Modern Information Systems".

MULTIOBJECTIVE OPTIMIZATION, HEURISTICS, GENETIC ALGORITHM, PARETO FRONT, SCHEDULE PROBLEM

## ЗМІСТ

<b>ВСТУП .....</b>	<b>9</b>
<b>1 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ СИСТЕМИ НАРАХУВАННЯ БОНУСІВ СПІВРОБІТНИКАМ КОМПАНІЇ .....</b>	<b>12</b>
1.1 Огляд сучасних досліджень в області впровадження систем мотивування на підприємствах .....	12
1.2 Опис бізнес-процесів .....	15
1.3 Опис постановки задачі .....	18
1.4 Огляд аналогів системи .....	18
Висновок до розділу .....	22
<b>2 МОДЕЛІ ТА МЕТОДИ ПІДТРИМКИ РОБОТИ СИСТЕМИ НАРАХУВАННЯ БОНУСІВ СПІВРОБІТНИКАМ КОМПАНІЇ .....</b>	<b>23</b>
2.1 Змістовна постановка задачі .....	23
2.2 Математична постановка задачі .....	24
2.3 Опис методів розв’язання задач .....	25
2.3.1 Опис методів розв’язання задач мінімізації загального часу завершення виконання робіт .....	25
2.3.2 Опис методів розв’язання задач багатокритеріальної оптимізації .....	30
2.3.2.1 Підхід з використанням згортки .....	32
2.3.2.2 Лексикографічний підхід .....	33
2.3.2.3 Підхід Парето .....	34
2.4 Опис алгоритму багатокритеріальної задачі складання розкладу .....	36
2.5 Експериментальні дослідження .....	43
<b>3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ...</b>	<b>46</b>
3.1 Вимоги до програмного забезпечення .....	46
3.2 Засоби розробки .....	49
3.3 Архітектура програмного забезпечення .....	51
3.4 Інструкція користувача .....	55
3.5 Опис технічного забезпечення .....	60
<b>4 РОЗРОБКА СТАРТАП-ПРОЕКТУ .....</b>	<b>62</b>



<b>4.1</b>	<b>Опис ідеї проекту .....</b>	<b>62</b>
<b>4.2</b>	<b>Опис конкурентів .....</b>	<b>63</b>
<b>4.3</b>	<b>Технологічний аудит ідеї проекту .....</b>	<b>65</b>
<b>4.4</b>	<b>Аналіз ринкових можливостей запуску стартап-проекту .....</b>	<b>67</b>
<b>4.5</b>	<b>Розроблення ринкової стратегії проекту .....</b>	<b>74</b>
<b>4.6</b>	<b>Розроблення маркетингової програми стартап-проекту .....</b>	<b>77</b>
	<b>Висновок до розділу .....</b>	<b>81</b>
	<b>ВИСНОВКИ .....</b>	<b>82</b>
	<b>СПИСОК ПОСИЛАНЬ .....</b>	<b>83</b>
	<b>ДОДАТОК А Графічний матеріал .....</b>	<b>89</b>
	<b>ПЛАКАТ 1 Схема структурна класів системи .....</b>	<b>90</b>
	<b>ПЛАКАТ 2 Схема структурна пакетів системи .....</b>	<b>91</b>
	<b>ПЛАКАТ 3 Схема структурна бази даних .....</b>	<b>92</b>
	<b>ПЛАКАТ 4 Графіки зміни множини рішень в процесі роботи алгоритму .....</b>	<b>93</b>
	<b>ПЛАКАТ 5 Екранні форми інформаційної системи .....</b>	<b>94</b>
	<b>ПЛАКАТ 6 Схема структурна варіантів використання .....</b>	<b>95</b>
	<b>ПЛАКАТ 7 Блок-схема багатокритеріального генетичного алгоритму .....</b>	<b>96</b>

## ВСТУП

Досвідчений керівник знає, що уважний підхід до співробітників, своєчасне заохочення і реагування на промахи - запорука успішного ведення бізнесу. Радянські часи запам'яталися дошками пошани, заохочувальними грамотами і сувенірними подарунками. Сьогодні способи мотивації бізнес-команди відрізняються від радянських.

Управління персоналом - окрема наука, в якій особливою увагою користуються методи мотивації персоналу.

Під мотивацією праці розуміють організацію роботи компанії таким чином, щоб кожен намагався виконувати професійні обов'язки найкращим чином і «викладався на повну». Іншими словами, кожен службовець отримує внутрішній стимул, який підвищує продуктивність праці і спрямований на досягнення спільної мети.

Часто адміністрація фірми використовує старі методи, що вже давно відійшли в минуле: заохочує кращих, карає гірших і проводить щорічні корпоративи для всіх. Але така схема працює не завжди. Навряд чи захочеться йти на корпоратив, якщо щоденна робоча обстановка пригнічує.

Грамотне стимулювання персоналу вимагає від керівника знання базових методів мотивації. Підлеглий, внесок якого в спільне досягнення не помітили або не оцінили, не захоче «проявляти себе» надалі [4].

Похвалу потребує кожен: і новачок, якому непросто «влитися» в колектив, і строгий кар'єрист, який під сухою маскою відповідальності і відданості компанії ховає бажання почути похвалу на свою адресу або отримати додаткову відпустку.

Керівник організації розробляє свою індивідуальну схему мотивації підлеглих співробітників.

Популярні методики мотивації персоналу розроблені на базі теорій мотивації [5]. Всього їх чотири:

- теорія Герцберга;
- теорія Тейлора;
- теорія Макклелланда;
- теорія Маслоу.

Герцберг пропонує скористатися зовнішніми і внутрішніми методами підвищення результативності роботи підприємства. Зовнішні методи передбачають створення комфортних умов праці, а внутрішні - виникнення у службовця задоволення від роботи в організації.

Тейлор радить використовувати для стимулювання підлеглих їх інстинкти і бажання задовольнити фізіологічні потреби. Він пропонує використовувати такі методи:

- оплату в залежності від виробітку або часу роботи;
- тиск;
- встановлення мінімальних норм виробітку;
- чіткі правила виконання функціональних обов'язків.

Макклелланд висуває теорію про види бажань людини:

- влада;
- успішність;
- приналежність до певної касты.

Основна частина службовців рада бути в рядах певної фірми і цінує свій статус. Лідери намагаються забрати владу, а одинаки працюють на особистий результат. Якщо умовно розділити співробітників за трьома категоріями, легко знайти підхід до кожного.

Маслоу пропонує розглядати роботу як можливість персоналу задовольнити свої потреби. Всі потреби Маслоу розглядає у вигляді ієрархії (піраміди). Він виділяє:

- фізіологічні потреби: їжа, вода, все, що потрібно для виживання;

- безпека: почуття захищеності необхідно для підтримки певного рівня життя;
- любов: бажання подобатися колегам;
- визнання: людина хоче мати певний статус в колективі і суспільстві в цілому;
- самовдосконалення.

Незважаючи на існування переліку розроблених схем мотивації персоналу, що прекрасно себе зарекомендували, багато керівників підприємств стикаються з деякими труднощами.

Сучасному керівнику потрібно грамотно підійти до питання мотивування підлеглих, щоб з найменшими витратами досягти високих показників функціонування організації.

Система мотивації повинна відповідати основним потребам колективу і легко модернізуватися слідом за мінливими умовами.

# **1 ПРОЕКТНІ РІШЕННЯ З РОЗРОБКИ СИСТЕМИ НАРАХУВАННЯ БОНУСІВ СПІВРОБІТНИКАМ КОМПАНІЇ**

## **1.1 Огляд сучасних досліджень в області впровадження систем мотивування на підприємствах**

Розвитку теорії і практики мотивації трудової діяльності присвячено багато книг та наукових досліджень як вітчизняних науковців, так і закордонних вчених таких як Л.М. Гохберг, С.Д. Ілленкова, Є.А. Добролюбов.

У статті [6] наведено огляд основних методів мотивації персоналу, які застосовуються на сучасних підприємствах. Доводиться, що прогресивна та інноваційна система мотивації праці має бути комплексною, тобто містити матеріальні та нематеріальні методи стимулювання. Також наведені рекомендації щодо розробки та впровадження методів мотивації персоналу, які можуть бути застосовані в системах стратегічного управління підприємствами. Зокрема, рекомендується взяти до уваги японську та французьку моделі мотивації персоналу, які, на думку авторів, враховують ринкові реалії сьогодення та українську традицію.

Японська модель мотивації персоналу базується на трьох факторах: професійна майстерність, вік та стаж роботи. В Японії неможливе автоматичне підвищення заробітної платні без підвищення кваліфікації працівника, що добре стимулює останнього розвиватись.

Французька модель мотивації персоналу містить дві складові, які формують підсумкову заробітну платню працівника – це індексація заробітної плати відповідно до зміни вартості життя в країні та врахування індивідуального внеску кожного працюючого. Індивідуальна складова базується на професійній кваліфікації, якості роботи, що виконується, мобільності працівника та кількості внесених раціоналізаторських пропозицій.

Отже, надається можливість не лише розкривати потенціал кожного працівника, а й одночасно вирішити задачі підприємства.

У роботі [7] досліджуються окремі теоретичні та практичні аспекти мотивації праці персоналу. Аналізується дія механізму матеріальної та нематеріальної мотивації трудової діяльності, пропонуються рішення щодо її поліпшення.

Також виведено 7 принципів, які слід враховувати при формуванні системи мотивації на підприємстві:

- розмір винагороди має визначатися на основі об'єктивної оцінки результатів праці працівника;
- зв'язок винагороди з результатами особистої праці – працівник має знати, яку винагороду він отримає за результати своєї праці;
- зв'язок винагороди з результатами діяльності колективу – винагорода має бути адекватною трудовому внеску кожного працівника в результат діяльності всього колективу;
- терміновість отримання винагороди – винагорода має бути вручена одразу за досягненням очікуваного результату;
- значущість винагороди – винагорода має задовольнити певну частку потреб працівника;
- зрозумілість винагороди – порядок визначення винагороди має бути зрозумілим кожному працівнику підприємства і бути з його позицій справедливими;
- стимулююча роль винагороди – винагорода має стимулювати подальшу активізацію професійної діяльності працівника, зростання її ефективності.

У статті [8] наведено теорії мотивації, їх характеристики та ілюстративне зображення співвідношення груп потреб у змістових теоріях

мотивації. Виходячи з даних теорій, описано 4 групи провідних мотивів працівників підприємств:

1) інтелектуальні мотиви. До цієї групи можна віднести інтелектуальний саморозвиток, захопленість ідеєю, прагнення до вирішення важких завдань та повної реалізації своїх інтелектуальних здібностей;

2) матеріальні мотиви – одержання матеріальної винагороди (заробітна плата, премії, надбавки);

3) моральні мотиви. Прикладом таких мотивів є визнання підприємством цінності роботи, внеску співробітника у розвиток підприємства;

4) статусні мотиви пов'язані з прагненням людини посісти вищу посаду, виконувати складнішу, відповідальнішу роботу, працювати у сфері діяльності (організації), яка вважається престижною, суспільно значущою.

Поняття мотивації об'єктивно знаходиться на перехресті двох наук – економіки та психології, саме тому при розробці стратегій по підвищенню мотивації на підприємстві слід враховувати як розробки психологів, так і індивідуальні потреби та характеристики особистості.

У статті [9] проаналізовано та виділено останні тенденції розвитку сфери матеріального мотивування в Україні. Досліджено нарахування бонусів залежно від рівня посади і визначено, що бонуси вищого керівництва найбільше залежать від результатів роботи компаній (75 %). Розміри бонусів менеджерів і спеціалістів, так само як і групи службовців/робітників, найбільше залежать від індивідуальних кількісних показників (76, 78 та 62 % відповідно). Отже, система матеріального стимулювання персоналу дозволяє врахувати й заохотити ряд додаткових кількісних і якісних характеристик праці, не охоплених основною заробітною платою. Ця система створює стимули тривалої дії.

У статті [10] розглянуто теоретичні аспекти системи мотивації праці як основного елемента підвищення конкурентоспроможності підприємства.

Подано концепцію стратегічного управління Balanced Scorecard (BSC) у процесі мотивації праці. Розглядається актуальність застосування технології «Key Performance Indicators», як елемента досягнення цілей підприємства. Також, запропоновано використання ключових показників ефективності при оцінці мотиваційної системи на підприємстві.

Зрозуміло, що тема впровадження мотиваційної системи на підприємствах є актуальною і активно досліджується.

## **1.2 Опис бізнес-процесів**

В кожній компанії рано чи пізно постає питання мотивації співробітників та задача розробки схеми KPI (ключових показників ефективності), виконання яких зазвичай призводить до преміювання [11].

Якщо взяти відділ продажів, то незалежно від специфіки сфери, найтипівіша модель фінансових відносин зводиться до фіксованої частини та бонусної, яка в більшості випадків пов'язана із відсотком від продажів.

Так чи інакше, в мотивації продавців все достатньо просто – чим більше продав, тим більше заробив.

Зовсім інакша ситуація складається з тими працівниками, які не впливають на прибутки компанії напряму. Наприклад, працівники служби технічної підтримки [12].

У будь-якій службі підтримки на верхньому рівні можна визначити 3 ключові ролі:

- оператор;
- спеціаліст 2-ї лінії;
- керівник команди технічної підтримки.

Розглянемо основні KPI для виконавців заявок, звернень, тобто працівників 2-ї лінії служби підтримки, які найчастіше використовуються для побудови моделі ключових показників та мотивації:

- кількість або відсоток виконаних заявок без запізнень;



- відсоток виконаних запитів з першого разу, тобто без отримання повторного запиту від клієнта;
- середня оцінка від клієнта за виконані звернення;
- час обробки звернень, трудовитрати;
- пріоритетність звернень від клієнтів;
- вартість виконаних заявок.

Враховуючи всі вищенаведені показники, необхідно створити баланс у виконанні заявок від клієнтів, застосовуючи правильну стратегію нарахування бонусів.

Наприклад, якщо для всіх звернень, заявок, завдань встановити фіксовану кількість бонусів, то можна стикнутися із такою проблемою, що працівники будуть обирати завжди найлегші заявки, а складні постійно будуть лишатися необробленими. Якщо кількість бонусів за виконане звернення, заявку, завдання встановити пропорційним його складності, то можна стикнутися з проблемою, що працівники навмисне будуть затягувати час виконання заявок, оскільки багато систем працює за таким принципом, що, чим більше часу витрачено на виконання, тим складнішою було завдання. Те ж саме стосується і встановленню кількості бонусів відповідно до пріоритетності. В цьому випадку існує така загроза, що низькопріоритетні звернення ніколи не будуть виконані.

В більшості компаній працівники самостійно обирають і розподіляють між собою звернення від клієнтів із загального пулу звернень, що дуже часто призводить до застосування «хитрощів» зі сторони працівників, згаданих вище. Або ж заявка просто призначається оператором на найменш завантаженого працівника по кількості заявок, не враховуючи їх складність.

Деякі компанії використовують технології штучного інтелекту, для обчислення кількості бонусів за виконані заявки. В таких випадках є

важливими такі параметри звернень, як кількість затраченого часу на обробку, хто є клієнт, тип звернення, встановлені кінцеві терміни для закриття заявки, оцінка роботи від клієнта і т.д. Та в більшості ситуацій такий підхід не є достатньо прозорим і штучний комп'ютерний інтелект не завжди в змозі коректно визначити оцінку виконаному завданню, на яке мають вплив дуже багато факторів. Тому, працівнику не зовсім зрозуміло, яким чином обчислюється кількість бонусів, що впливає на зниження мотивації. Також, витрати коштів бюджету компанії для бонусних виплат використовуються не оптимальним чином.

Отже, пропонується розглянути новий підхід до впровадження бонусної системи працівникам компанії, зокрема відділу технічної підтримки.

Кожне звернення, заявка від клієнта мають певну заплановану базову кількість бонусів. В залежності від того, який працівник буде виконувати цю заявку, базова кількість бонусів може змінюватися. Так, в командах зазвичай працюють спеціалісти різних рівнів. Відповідно, час виконання заявки буде залежати від рівня досвідченості працівника. Так, сеньйор може виконати те саме завдання в рази швидше, ніж молодий малодосвідчений джуніор. Таким чином, досвідченіші працівники завершують завдання значно раніше запланованого та отримують при цьому більшу кількість бонусів.

Отже, весь пул заявок розподіляється між працівниками таким чином, щоб мінімізувати загальний час завершення виконання завдань та витратити при цьому мінімальну суму бонусів. Тепер кожен працівник має власну призначену множину заявок, і йому більше не потрібно домовлятися та розподіляти завдання разом з іншими колегами. Критичні звернення виконуються позачергово, будь-яким найменш завантаженим членом команди.

Отже, усувається людський фактор при розподілі завдань у команді. Оптимізуються трудозатрати команди. Ресурси компанії витрачаються оптимальним шляхом.

### **1.3 Опис постановки задачі**

Необхідно розробити систему для підтримки процесу нарахування бонусів співробітникам компанії як складової мотиваційної програми. Система повинна забезпечувати виконання таких завдань:

- збереження даних по працівникам;
- збереження даних, що стосуються завдань, які необхідно виконати;
- призначення кожному завданню еталонної кількості бонусів;
- на основі даних про працівників та даних про завдання знайти оптимальні розподіли завдань між працівниками таким чином, щоб мінімізувати загальний час завершення виконання завдань та загальну вартість винагород;
- надати користувачу можливість вибору підходящого розподілу завдань між працівниками із множини потенційних оптимальних рішень;
- збереження даних про розподіл завдань між працівниками;
- збереження даних про кількість бонусів, отриманих кожним працівником.

Схема структурна варіантів використання наведена в додатку А.

### **1.4 Огляд аналогів системи**

Розглянемо аналоги систем, що використовуються для нарахування бонусів співробітникам компанії, що вже існують на ринку.

**SalesupCRM.** SalesupCRM – це хмарна CRM-система для обліку клієнтів та угод, яка допомагає навести порядок у відділі продажів та

підвищити прибутки. Програма зберігає всю історію спілкування із клієнтами та записи дзвінків, полегшує контроль за співробітниками, автоматизує рутинні процеси, допомагає управляти задачами і проектами і генерувати звіти у реальному часі.

Система має розділ «Нарахування» для ведення обліку виплат заробітних плат та бонусів співробітникам. Бонуси за продажі можна нараховувати вручну, перейшовши до розділу Співробітники -> Нарахування. На формі можна обрати потрібного співробітника, ввести суму, тип та період нарахування. Якщо необхідно, можна вказати також за яку угоду або платіж нараховується бонус. Приклад форми наведено на рисунку 1.1.

**Новое начисление**

Сотрудник: Ксения

Тип начисления: %

Тип объекта: Заявка

Объект: ★ Поставка на 230 000 Р №...

Сумма: 5000

Статус: ● Подтверждено

Период: 07.2019

Основание: Проценты за продажу

Комментарий

Создать

2 начисления на 100 125 Р    Оплачено 50 000 Р

Сумма	Создано	Тип начисления	Основание	Статус	Статус оплаты	Комме
100 000 Р	09.10.19 09:55	Аванс	...	● Не назначено	Частично оплачен	...
125 Р	15.05.19 11:11	%	4	● Подтверждено	Не оплачен	...

к 3.8.75 4767-6358

Рисунок 1.1 – Приклад форми нарахування бонусів співробітнику за вкладену угоду

Також нарахування бонусів можна автоматизувати за допомогою сценаріїв автоматизації. Наприклад, нараховувати 5% від суми кожної закритої угоди або 100\$ за угоду, прибуток від якої перевищує 10000\$ [13].

Недоліком даної системи є те, що вона обмежена у функціональності нарахування бонусів співробітникам та розрахована лише для працівників відділу продажу.

**Compensation for Creatio.** Compensation for Creatio – це доповнення до системи, що дає можливість вести облік бонусів та мотивації співробітників, а також контролювати виплати та нарахування. Дане застосування розроблено для компаній, які мають складну систему нарахування бонусів і винагород та існує необхідність контролювати даний процес встановленими правилами нарахування для співробітників та команд [14].

Застосування має такі можливості:

- мотивація співробітників за допомогою прозорості системи бонусних виплат;
- налаштування правил розрахунків бонусів, умов, частоти та способу їх нарахування;
- нарахування бонусів не лише окремим співробітникам, але й цілим командам;
- налаштування процесу візування для контролю виплат;
- відслідковування потрібних показників та статистичних даних по бонусних виплатах за допомогою зручних графіків.

Приклад однієї із форм системи наведено на рисунку 1.2.

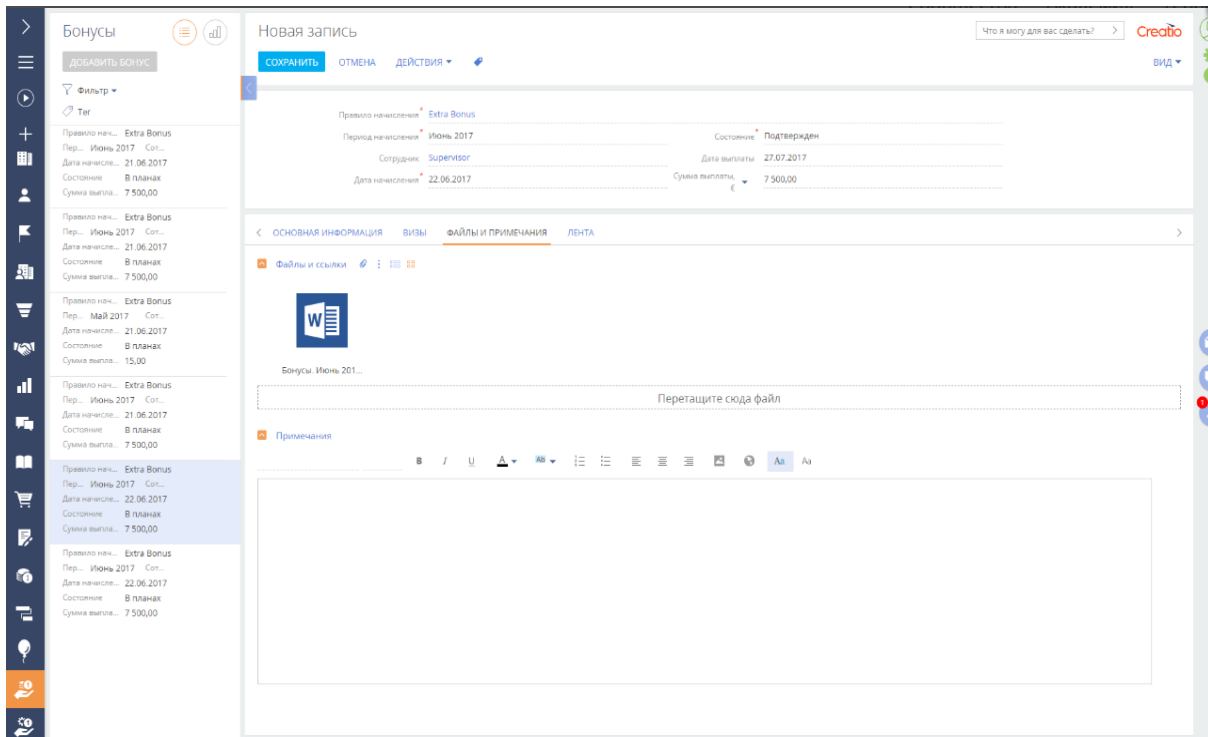


Рисунок 1.2 – Приклад створення нового запису про бонуси у системі Creatio за допомогою додатку «Compensation for Creatio»

Недоліком системи є те, що бонуси нараховуються на основі створених правил. У реальному житті об'єктивно визначити суму винагороди за виконання завдання, спираючись на декілька створених правил, дуже важко, оскільки існує дуже багато факторів, які важко описати за допомогою правил. Відсутній інструмент для оптимізації бонусних витрат.

**Bonus Management System.** Автоматизоване рішення для розрахунку та нарахування премій на базі системи управління персоналом БОС-Кадровик має наступні можливості:

- визначення періодів преміювання;
- наявність різних шкал для розрахунку бонусів;
- розрахунки КРІ на основі категорій задач співробітників;
- формування типових та індивідуальних бонусних схем;
- на основі наведених даних розрахунок та нарахування колективних, групових та персональних премій.

В основу процесу закладено вибір періодів преміювання та формування типових бонусних схем, які відповідають прийнятим на підприємстві матрицям розподілу бонусних винагород [15].

Недоліком даної системи є те, що вона розрахована для проведення лише загальних бонусних винагород в певні періоди року.

### **Висновок до розділу**

В даному розділі розглянуто різні дослідження в області мотивування співробітників компанії. Наведено 7 основних принципів, які слід враховувати при впровадженні мотиваційної програми на підприємстві та описано 4 провідних мотиви працівників підприємств. Було проаналізовано дослідження, що показує, що бонуси керівництва компанії найбільше залежить від результатів роботи компанії, тоді як бонуси звичайних працівників найбільше залежать від індивідуальних кількісних показників. Було розглянуто бізнес-процес нарахування бонусів на прикладі роботи відділу технічної підтримки компанії. Було сформульовано постановку задачі. Також розглянуто аналоги системи, що розробляється, та визначено недоліки кожної з них.

## **2 МОДЕЛІ ТА МЕТОДИ ПІДТРИМКИ РОБОТИ СИСТЕМИ НАРАХУВАННЯ БОНУСІВ СПІВРОБІТНИКАМ КОМПАНІЇ**

### **2.1 Змістовна постановка задачі**

Продуктова компанія N має власну службу технічної підтримки, яка допомагає клієнтам вирішувати всі питання, що виникають в процесі роботи з продуктом компанії. Це можуть бути консультації по використанню функціональності, допомога з налаштуванням системи, виправлення помилок в роботі системи. Кожного дня команда отримує множину звернень від клієнтів.

Команда складається з 6 чоловік:

- оператор технічної підтримки;
- керівник команди;
- бізнес аналітики 2-ї лінії.

Оператор лише приймає звернення від клієнтів, а бізнес-аналітики 2-ї лінії та їх керівник працюють над вирішенням отриманих звернень. Серед працівників є спеціалісти різних рівнів – джуніори, мідли та сеньйори.

Працівники технічної підтримки мають постійну складову матеріальної винагороди – заробітну плату та змінну – бонуси, що залежать від результатів виконаної роботи. Кожне звернення оцінюється базовою кількістю бонусів і в залежності від рівня досвідченості працівника кількість бонусів може змінюватися.

Необхідно розподілити всі заявки, що надійшли від клієнтів, між членами команди так, щоб мінімізувати загальні витрати на бонуси та одночасно досягти високої продуктивності виконання робіт.



## 2.2 Математична постановка задачі

Нехай задано множину робіт  $J = \{1, 2, \dots, j, \dots, n\}$  і множину виконавців  $M = \{1, 2, \dots, i, \dots, m\}$ . Кожна робота може виконуватися будь-яким виконавцем. Передбачається, що виконавці мають різну швидкість виконання робіт  $v_i$ ,  $v_i > 0, i = \overline{1, m}$ .

Для  $j$ -ї роботи задано час виконання  $p_j$  та кількість бонусів  $b_j > 0$ ,  $j = \overline{1, n}$ . Якщо  $v_i > 0$  – швидкість  $i$ -го виконавця, то тривалість виконання  $j$ -ї роботи  $i$ -м виконавцем становить:

$$p_{ij} = \frac{p_j}{v_i}, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Кількість бонусів, нарахованих  $i$ -му виконавцю за  $j$ -ту роботу, залежить від швидкості роботи цього виконавця і обчислюється за формулою:

$$b_{ij} = \frac{b_j}{v_i}, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Загальну суму нарахованих бонусів за увесь обсяг робіт отримуємо з:

$$B = \sum_{i=1}^m \sum_{j=1}^n a_{ij} b_{ij} = \sum_{i=1}^m \frac{1}{v_i} \sum_{j=1}^n a_{ij} b_j,$$

де  $a_{ij} \in \{0, 1\}$ ,  $a_{ij} = 1$ , якщо  $j$ -та робота призначена  $i$ -му виконавцю.

Нехай маємо деякий розклад виконання робіт. Позначимо через  $C_{ij}$  – момент завершення  $j$ -ї роботи  $i$ -м виконавцем. Загальний час виконання робіт визначається наступним чином:

$$C_{\max} = \max_{ij} \{C_{ij}\}.$$

Необхідно скласти розклад виконання робіт, при якому загальний час завершення робіт множини  $J$  буде мінімальним і при цьому загальна кількість нарахованих бонусів також буде мінімальною:

$$C_{\max} \rightarrow \min, \quad (2.1)$$

$$B \rightarrow \min. \quad (2.2)$$

## 2.3 Опис методів розв'язання задач

В даному розділі розглянемо методи розв'язання задач мінімізації загального часу завершення виконання робіт, що виступає одним із критеріїв поставленої задачі, та методи вирішення задач багатокритеріальної оптимізації.

### 2.3.1 Опис методів розв'язання задач мінімізації загального часу завершення виконання робіт

Для розв'язання поставленої задачі необхідно використовувати моделі та методи теорії розкладів [16].

Проблема побудови та оптимізації розкладів належить до класу NP-повних. Для таких категорій задач характерною є відсутність алгоритмів розв'язку за поліноміальний час залежно від розмірності.

З точки зору точності розв'язків задач теорії розкладів [17], методи можна розділити на точні та наближені. Найпопулярнішими є наступні методи:

а) точні методи:

1) методи цілочисельного програмування:

- лінійне цілочисельне програмування;
- нелінійне цілочисельне програмування; булеве програмування;

2) послідовні алгоритми оптимізації:

- метод гілок та меж;

- динамічне програмування та методи аналізу і відсіву варіантів;
- методи теорії графів;

b) наближені методи:

1) обмеження об'єму розрахунків у послідовних алгоритмах оптимізації;

2) методи випадкового пошуку:

- методи глобального випадкового пошуку,
- методи локальної варіації;

3) генетичні алгоритми та еволюційні стратегії;

4) евристичні методи;

5) гібридні алгоритми;

6) метаевристичні методи;

7) імітаційне моделювання.

Евристичні методи розв'язання задач користуються великою популярністю. Евристичні методи дозволяють знаходити прийнятні рішення навіть у дуже складних випадках: при неповноті, випадковості вхідних даних, відсутності адекватної математичної моделі, NP-складності вирішуваної задачі та відсутності точних методів їх рішення.

Більшість задач теорії розкладів є NP-важкі, але реальні практичні задачі потребують їх вирішення. Для цього використовується декілька підходів. Першим підходом є розробка поліноміальних евристичних алгоритмів. Евристичний алгоритм, або просто евристика — це алгоритм, спроможний видати прийнятне рішення проблеми серед багатьох рішень, але неспроможний гарантувати, що це рішення буде найкращим [18]. Отже, такі алгоритми є приблизними і неточними. Зазвичай такі алгоритми знаходять рішення, близьке до найкращого і роблять це швидко [19-20]. Іноді такий алгоритм може бути точним, тобто він

знаходить дійсно найкраще рішення, але він все одно буде називатися евристичним доти, доки не буде доведено, що рішення дійсно найкраще.

Серед евристичних алгоритмів значне місце посідають жадібні алгоритми [21]. Це простий і прямолінійний евристичний алгоритм, який приймає найкраще рішення, виходячи з наявних на поточному етапі даних, не турбуючись про можливі наслідки, сподіваючись врешті-решт отримати оптимальне рішення [22]. Легкий в реалізації і часто дуже ефективний за часом виконання. Недоліком є те, що багато задач не можуть бути розв'язані за його допомогою.

Генетичні алгоритми [23 – 25] дозволяють розв'язувати широке коло складних задач багатокритеріальної оптимізації шляхом випадкового підбору, комбінування та зміни параметрів моделювання способами, що подібні до біологічної еволюції (наслідування, мутація, відбір). Застосовують ці алгоритми й для розв'язку задач на графах. Генетичні алгоритми оперують сукупністю особин (популяцією), які кодують один із можливих розв'язків задачі. Цим генетичний алгоритм відрізняється від більшості інших алгоритмів оптимізації, які оперують лише з одним розв'язком, який покращує його [26 – 27]. Принцип роботи генетичного алгоритму полягає у наступному. За допомогою функції пристосованості серед всіх особин популяції виділяють:

- найбільш пристосовані, які отримують можливість схрещуватися і давати потомство;
- найгірші, які видаляються з популяції і не дають потомства.

Таким чином, пристосованість нового покоління в середньому вище попереднього.

Розмірність задачі напряду збільшить довжину «генома», але можливість виконати задану кількість ітерацій може дозволити отримати наближене до оптимального або допустиме непротирічне рішення;

Недоліки алгоритму: час виконання функції оптимізації, необхідність знайти всі рішення задачі, а не одне з них, багато тимчасових даних, недоведеність збіжності.

Алгоритми імітації відпалу [28] – загальний алгоритмічний метод розв'язання задачі глобальної оптимізації; є одним із прикладів методів Монте Карло. У процесі пошуку оптимального рішення з деякою ймовірністю допускається перехід у стан із більш високим значенням цільової функції. Ця властивість дозволяє їм виходити з локальних оптимумів. На початку роботи алгоритму ймовірність переходу в стан із більш високим значенням цільової функції повинна бути досить велика, щоб була можливість переходу від обраного початкового наближення до будь-якого іншого розв'язання. У процесі роботи алгоритму ймовірність переходу поступово зменшується.

Методи імітації відпалу характеризуються великою обчислювальною складністю при великій розмірності.

Метод гілок та меж [29] є загальним алгоритмічним методом вирішення різноманітних оптимізаційних задач. Даний метод є варіацією повного перебору, при якому відкидаються підмножини допустимих рішень, що не містять оптимальних значень цільової функції. В методі гілок та меж використовуються дві процедури: розгалуження та знаходження оцінок (меж). За допомогою процедури розгалуження множина допустимих рішень на кожному етапі розділяється на підмножини меншої розмірності. Ці множини стають вузлами дерева пошуку. Процедура знаходження оцінок (меж) визначає верхні та нижні межі оптимального значення на підмножинах допустимих рішень. Якщо для знайденого вузла дерева пошуку верхня межа співпадає з нижньою, то це значення цільової функції є оптимальним і досягається на відповідній підмножині допустимих рішень.

Метод розфарбування графу. Складання розкладу можна представити як процес розфарбовування графу – пошук мінімального числа кольорів, необхідних для розфарбовування вершин деякого графу так, щоб для кожної пари сусідніх вершин використовували різні кольори.

Маємо задачу теорії розкладів з незалежними паралельними «машинами». Різні методи розв'язання даної задачі широко досліджуються в літературі останні роки.

В статті [30] розглянута проблема складання розкладу для паралельних машин для зменшення загального часу виконання. Для вирішення поставленої проблеми використовується меметичний алгоритм. Також приводиться розрахунок та порівняння складності меметичного алгоритму та звичайного генетичного алгоритму.

Задача складання розкладу для паралельних машин розглянута в статті [31]. Для вирішення поставленої задачі використовується генетичний алгоритм з перестановками.

У статті [32] розглянута задача складання розкладу для непов'язаних паралельних приладів з різним часом обробки робіт для мінімізації загального часу виконання. Для вирішення поставленої задачі використовується покращений мурашиний алгоритм.

У статті [33] пропонується реалізація Tabu Search алгоритму для вирішення задач планування для паралельних машин із часом встановлення, залежним від послідовності та машини, щоб мінімізувати максимальний час завершення робіт. Проблема непроста, а знайти оптимальне рішення ефективно малоімовірно. Тому евристичні прийоми є більш доречними для пошуку майже оптимальних рішень. Запропонований Tabu Search алгоритм використовує дві фази схем збурень: внутрішньомашинне збурення, що оптимізує послідовність завдань на машинах, і міжмашинне збурення, що врівноважує розподіл завдань машинам. Було проведено порівняння запропонованого

алгоритму з існуючими. Результати обчислень показали, що запропонований Tabu Search алгоритм ефективніший існуючої евристики для задач малої та великої розмірності.

Мета статті [34] - знайти оптимальний розклад (розв'язок) для задачі теорії розкладів з використанням ідентичних паралельних машин. В даній роботі розроблений та використаний алгоритм для вирішення проблеми. Для формулювання задачі використовується модель лінійного програмування ILP. Найбільше часу витрачається для алгоритму LPT, що використовується для пошуку (генерації) початкового рішення, потім використовується розроблений алгоритм для вдосконалення рішення. Результати досліджень продемонстрували, що математичне моделювання та алгоритми є потужним інструментом і є більш ефективним для таких видів задач, порівняно з традиційними методами.

У статті [35] розглянута задача складання розкладу для паралельних машин з можливими перериваннями. Для вирішення поставленої задачі автори статті пропонується використовувати апроксимаційний алгоритм із складністю  $O(Nn \log n)$ .

Отже, як видно з наведених статей, пошук ефективних алгоритмів для розв'язання задач теорії розкладів, що мають широке практичне застосування, є актуальним питанням.

### **2.3.2 Опис методів розв'язання задач багатокритеріальної оптимізації**

Так як маємо багатокритеріальну задачу оптимізації, то визначення оптимального розв'язку перетворюється у нетривіальну задачу. Об'єктивно не можна визначити найкраще рішення, якщо критеріїв багато і вони ще, можливо, між собою конфліктують. Тому необхідно шукати компромісне рішення, що враховує важливість кожної цільової функції [36].

Так як поставлена задача оптимізації складається із 2 цільових функцій – мінімізація загального часу завершення виконання робіт та мінімізація загальних бонусних витрат, то пошук одного чи більше оптимальних рішень називається багатокритеріальною оптимізацією, а задача називається задачею багатокритеріальної оптимізації [37-38].

Дана задача активно досліджується вченими всього світу. У роботі [39] наведено короткий огляд багатокритеріальної оптимізації з використанням метаевристики. Представлено опис всіх використаних алгоритмів. Не дивлячись на те, що основний наголос в даній роботі робиться на еволюційні алгоритми, такі метаевристики як оптимізація рою частинок, штучна імунна система та оптимізація мурашиних колон також досліджуються. У статті [40] аналізуються методи багатокритеріальних оптимізацій, наводиться їх формулювання, переваги та недоліки. Ефективність вибраних методів для цілей інженерного проектування визначається шляхом порівняння результатів, отриманих при рішенні тестових задач та реальної конструкторської інженерної задачі.

Робота [41] демонструє рішення задачі багатокритеріальної оптимізації за допомогою розробленої модифікації еволюційного алгоритму SPEA2.

Як видно з практики, евристичні методи широко застосовуються та мають переваги над класичними. Одним із найпопулярніших представників евристичних методів є генетичний алгоритм.

Існує три підходи до вирішення задач багатокритеріальної оптимізації.

**Підхід з використанням згортки.** Цей підхід базується на перетворенні задачі багатокритеріальної оптимізації в задачу однокритеріальної оптимізації за допомогою скаляризації векторного критерію в один узагальнений, при цьому ваги кожного критерію



визначаються користувачем. Це свого роду процес багатокритеріальної оптимізації, що базується на вподобаннях.

**Лексикографічний підхід.** В його основу покладено розташування критеріїв у порядку пріоритетів. Це знову ж таки підхід, що базується на вподобаннях.

**Підхід Парето.** Знаходить всі можливі розв'язки, що не домінуються, і повертає користувачу множину оптимальних розв'язків. Це свого роду ідеальний процес багатокритеріальної оптимізації.

### 2.3.2.1 Підхід з використанням згортки

Як тільки цільові функції (критерії) були визначені, цей метод призначає числове значення ваг кожній цільовій функції та потім комбінує значення зважених критеріїв в одну цільову функцію шляхом додавання або множення всіх зважених критеріїв, перетворюючи таким чином багатокритеріальну задачу оптимізації в однокритеріальну. Аналітично цільова функція даного підходу може бути виражена таким чином:

$$F(x) = \sum_{i=1}^n w_i F_i(x), \text{ або}$$

$$F(x) = \prod_{i=1}^n F_i(x)^{w_i},$$

де  $n$  - кількість критеріїв;

$w_i, i = \overline{1, n}$  - числове значення вагів  $i$ -го критерію.

Переваги такого підходу:

- простий для розуміння та використання.

Недоліки:

- вибір значень ваг є суб'єктивним та індивідуальним;
- втрачається можливість знайти інші корисні розв'язки, маючи один компроміс між критеріями;

- різні критерії з різними одиницями виміру вимагають відповідного процесу нормалізації, який часто є суб'єктивним;
- має проблему поєднання несумірних критеріїв;
- необхідність декількох запусків алгоритму, заснованого на даному підході для того, щоб отримати декілька компромісних розв'язків багатокритеріальної задачі.

### **2.3.2.2 Лексикографічний підхід**

Цей підхід працює наступним чином. Різні пріоритети призначаються різним критеріям, а потім критерії оптимізуються в порядку їх пріоритетів. Таким чином, коли два чи більше розв'язків порівнюються один з одним для того, щоб обрати найкращий, то їх показники ефективності порівнюються за критерієм, що має найвищий пріоритет. Якщо один розв'язок-кандидат краще іншого відповідно до цього критерію, то обирається саме він. Інакше, показники ефективності двох можливих розв'язків порівнюються відносно другого критерію. І знову ж, якщо розв'язок є кращим за другим критерієм, то він і обирається. Цей процес повторюється доти, поки не буде знайдено явного переможця або ж поки не будуть розглянуті всі критерії. В останньому випадку, якщо немає явного розв'язку-переможця, то можна обрати розв'язок, що оптимізує найбільш пріоритетний критерій.

Переваги даного підходу:

- уникає проблеми поєднання несумірних критеріїв в одній формулі, розглядає кожен критерій окремо;
- володіє концептуальною простотою та зручністю використання.

Недоліки:

- визначення порогу допуску та відповідного ступеня впевненості для кожного критерію суб'єктивне та довільне;
- послідовний за своєю природою.

### 2.3.2.3 Підхід Парето

Цей підхід отримав назву в честь інженера 19-го століття, економіста та соціолога В. Парето. Замість того, щоб перетворювати багатокритеріальну задачу оптимізації в однокритеріальну і потім вирішувати її за допомогою однокритеріального оптимізатора, використовується алгоритм багатокритеріальної оптимізації для вирішення оригінальної задачі. Замість одного оптимального розв'язку, як в задачах однокритеріальної оптимізації, даний підхід знаходить всі можливі оптимальні розв'язки і повертає набір недомінуючих оптимальних розв'язків, що називаються оптимальними розв'язками по Парето, або фронт Парето [42-43].

Таким чином, приходимо до поняття «ефективного» (оптимального по Парето) розв'язку. Властивістю ефективності повинно володіти кожне рішення, що претендує на звання оптимального.

Нехай  $X$  - множина допустимих розв'язків задачі,  $x \in X$  - допустимий розв'язок. Вважаємо, що кожний розв'язок  $x \in X$  оцінюється по  $n=2$  критеріям.

Нехай  $H_i(x), x \in X$  - функція, значеннями якої є оцінки розв'язку  $x \in X$  за критерієм  $i, i = \overline{1, n}$ . Тоді вектор  $H(x) = (H_1(x), \dots, H_n(x)), x \in X$  - набір оцінок розв'язку  $x \in X$  по всім критеріям. Розв'язок  $x^* \in X$  вважається оптимальним (оптимальним по Парето, ефективним), якщо не існує іншого розв'язку  $x \in X$  такого, що

$$H_i(x) \geq H_i(x^*), i = \overline{1, n},$$

$$\exists i_0 : H_{i_0}(x) > H_{i_0}(x^*).$$

Тобто, іншими словами, якщо  $x^* \in X$  - оптимальний по Парето розв'язок, то не існує іншого розв'язку  $x \in X$ , що є кращим за  $x^*$  хоча б по одному критерію та не гіршим по інших критеріях.

Множина всіх ефективних (оптимальних по Парето) розв'язків називається ефективною множиною.

Коли намагаємося оптимізувати декілька критеріїв одночасно, то простір пошуку також стає частково впорядкованим. Для отримання оптимального рішення буде набір оптимальних компромісів між конфліктуючими критеріями. Як показано нижче на рисунку 2.1, розв'язок може бути кращим, гіршим, а також нейтральним по відношенню до критеріїв (ні домінуючим, ні домінантним). Оптимальний розв'язок – це рішення, що не домінується будь-якими іншими рішеннями в просторі пошуку.

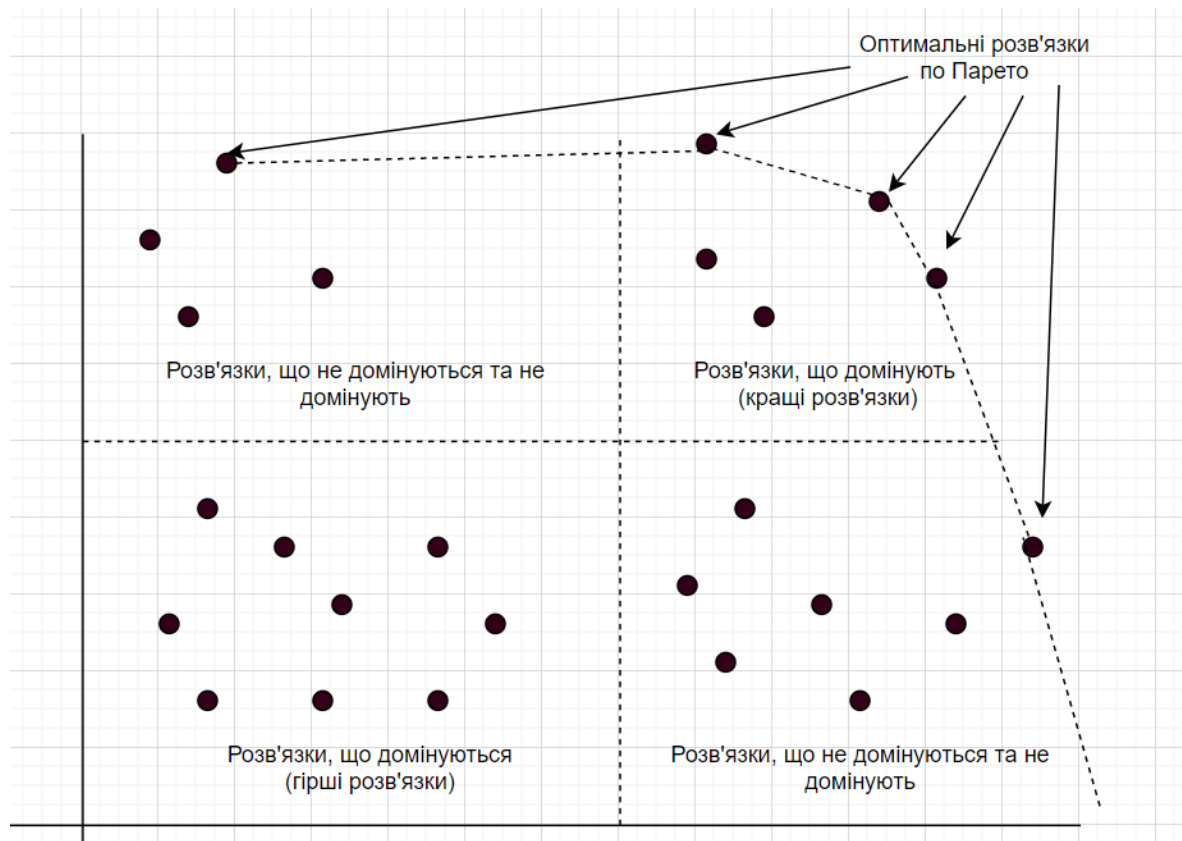


Рисунок 2.1 – Впорядкований простір пошуку розв'язків

Переваги даного підходу:

- ніколи не комбінує різні критерії в одній формулі, всі критерії розглядаються окремо;

- справляється із несумірними критеріями;
- результат у вигляді набору домінуючих розв’язків представлений у формі фронту Парето, що залишає можливість вибору користувачу підходящого оптимального розв’язку; уникає багаторазового запуску алгоритму.

Недоліки:

- набагато складніший, ніж інші два підходи.

Враховуючи переваги та недоліки всіх вищенаведених підходів, будемо використовувати підхід Парето.

## 2. 4 Опис алгоритму багатокритеріальної задачі складання розкладу

Для розв’язання багатокритеріальної задачі теорії розкладів для мінімізації загального часу завершення виконання робіт паралельними виконавцями, що мають різну швидкість роботи та мінімізації загальної кількості бонусів було обрано багатокритеріальний генетичний алгоритм. Схема роботи генетичного алгоритму наведено на рисунку 2.2.

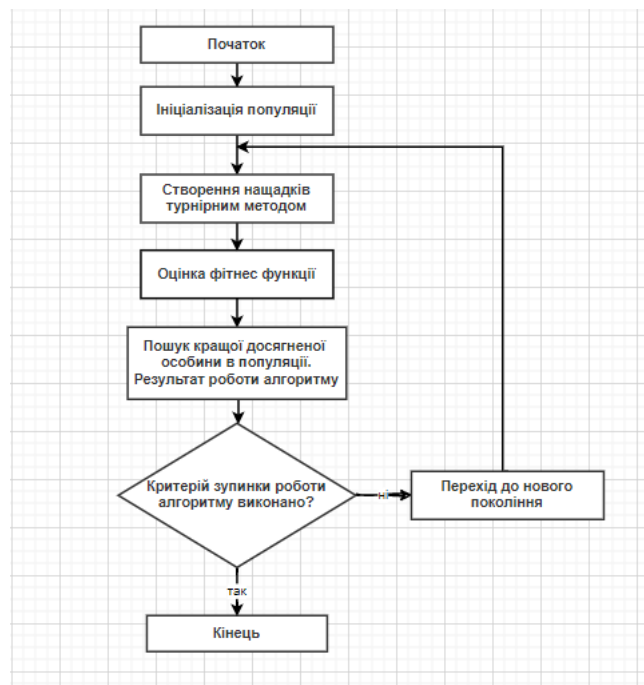


Рисунок 2.2 – Схема роботи генетичного алгоритму

Важливою перевагою генетичних алгоритмів є те, що вони з самого початку працюють з великою популяцією кандидатів на розв'язки, і, відповідно, оптимальні по Парето рішення генеруються вже в 1-му поколінні. Таким чином, не зважаючи на складність поставленої задачі, можна отримати достатньо повну множину оптимальних по Парето розв'язків уже після першої ітерації процедури генетичного алгоритму при відносно невеликих затратах часу на обчислення [44].

Модифікація генетичного алгоритму для розв'язання багатокритеріальної задачі складання розкладу відрізняється від звичайної схеми оператором селекції, визначення кращих особин популяції, обчисленням значення пристосованості особин та застосуванням підходу Парето.

Отже, для кожної особини популяції обчислюються окремо значення кожного з критеріїв: загальний час завершення виконання робіт в системі та загальна сума бонусів.

Після обчислення значень критеріїв застосовується підхід Парето. Для кожної особини популяції визначається ранг границі Парето. Особини (розклади), що знаходяться безпосередньо на границі мають ранг 1. Якщо видалимо ці особини із популяції та визначимо нову границю, то особини нової границі матимуть ранг 2 і так далі. Отже, ранги визначаються наступним чином: додаємо особину до границі, якщо вона не домінується іншими особинами, що вже перебувають на границі та видаляємо із границі всі особини, що домінуються новою особиною. Також для всіх особин кожного рангу обчислюється відстань до сусідніх особин (найближчих зліва та справа по значенню критерію (2.1)). Тоді для селекції особин поточної популяції та їх нащадків для створення нового покоління необхідно впорядкувати всіх особин по зростанню рангу і, після цього, по спаданню відстані до сусідніх особин. Таким чином до нового покоління потрапляють перші  $n$  особин. Відстань до сусідніх особин одного рангу необхідно

обчислювати для того, щоб уникнути збіжності до локального мінімуму та зберегти різноманітність осіб популяції (рисунк 2.3). Так, наприклад, позначені на рисунку 1 особини мають в 2 рази більше шансів бути обраними в якості батьків для створення нащадків, що може призвести до хибних результатів. Тому для наступного покоління обираються лише особини з великою відстанню до сусідніх особин.

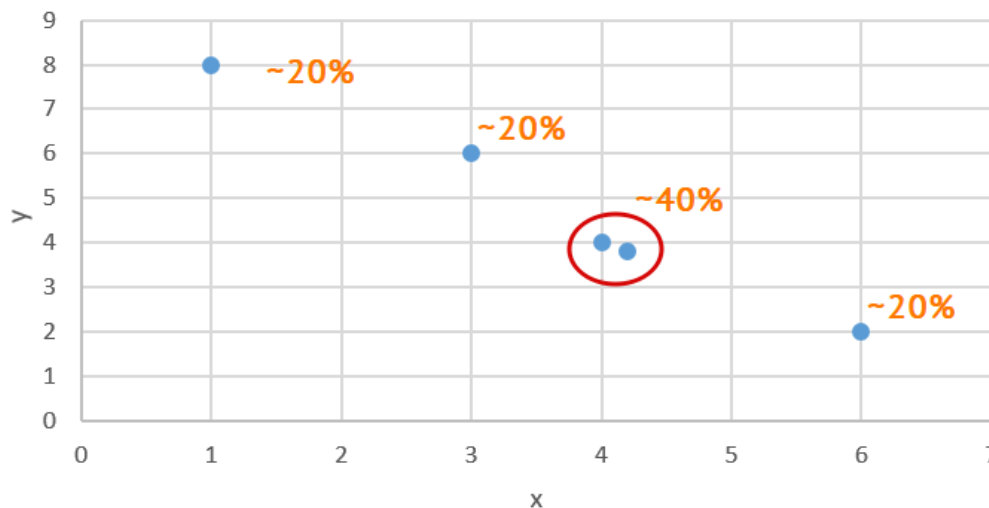


Рисунок 2.3 – Ілюстрація відстані між особинами та ймовірністю їх селекції для створення нащадків

Схема роботи генетичного алгоритму для розв'язання задачі багатокритеріальної оптимізації наведено нижче.

КРОК 1. Згенерувати початкову популяцію.

КРОК 2. Обробити популяцію.

КРОК 2.1. Створити нащадків, використовуючи турнірний метод для вибору батьків.

КРОК 2.2. Додати до поточного покоління всіх нащадків та нормалізувати значення фітнес-функцій – загальний час завершення робіт та загальна сума бонусів.

КРОК 2.3. Визначити ранги особин відповідно до рівнів домінування.

КРОК 2.3.1. Задати початкове значення рангу для всіх особин  $r_i = -1, i = \overline{1, N}$  та додати до множини перевірки на домінантність всіх особин популяції.

КРОК 2.3.2. Встановити початкове значення рангу  $R = 1$ .

КРОК 2.3.3. Для кожної особини множини перевірки на домінантність перевіряємо чи не домінується вона, тобто є кращою хоча б по одному критерію та не гіршою за іншими критеріями відносно інших особин із множини перевірки на домінантність.

КРОК 2.3.4. Для всіх особин, що не домінуються, присвоїти  $r_i = R$  та видалити їх із множини для перевірки на домінантність.

КРОК 2.3.5. Збільшити значення рангу  $R = R + 1$ .

КРОК 2.3.6. Якщо множина особин для перевірки на домінантність не порожня, то перейти на КРОК 2.3.3, інакше перейти на КРОК 2.4.

КРОК 2.4. Для особин кожного рангу обчислити відстань до сусідніх особин.

КРОК 2.4.1. Відстань обчислюється як сума відстаней до лівої та правої сусідньої особини. Для першої та останньої особини рангу відстань рівна нескінченності.

КРОК 2.5. Впорядкувати всіх особин популяції спочатку по рангу, а потім по спаданню відстаней між особами в межах кожного рангу.

КРОК 2.6. Створити нову популяцію, що складається з перших  $N$  особин впорядкованої множини.

КРОК 2.7. Якщо не досягнуто максимальної кількості популяцій, то перейти на КРОК 2, інакше перейти на КРОК 3.

КРОК 3. В якості множини розв'язків визначити множину особин першого рангу популяції – Парето фронт.



На рисунку 2.4 проілюстровано приклад визначення відстаней між особинами популяції одного рангу. Тут  $\overline{l_1, \dots, l_5}$  - це відстані між двома особинами,  $\overline{d_1, \dots, d_5}$  - сума відстаней від особини до сусідніх особин зліва та справа, при цьому для крайніх точок (особин) відстань до сусідніх особин визначається як  $\inf$  (дуже велике число).

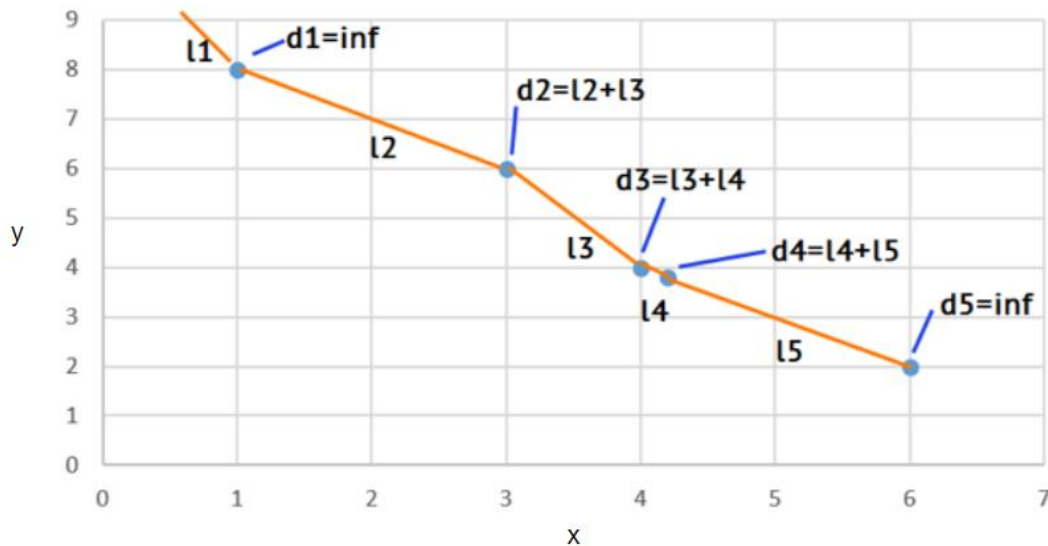
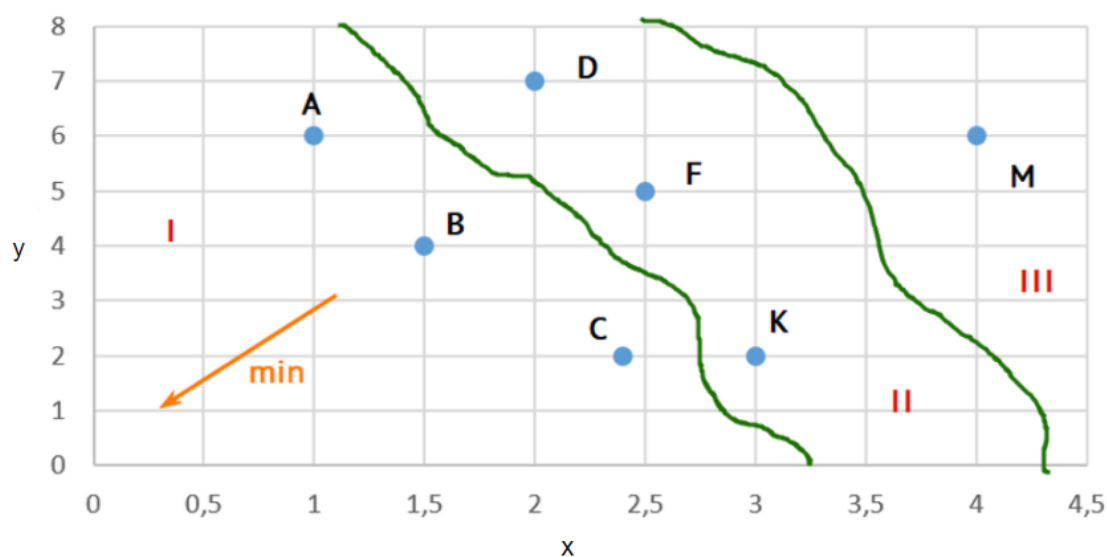


Рисунок 2.4 – Визначення відстаней між особинами одного рангу

На рисунку 2.5 наведено приклад визначення рангів для двокритеріальної задачі мінімізації. Точки А, В, С із області І не домінуються точками із І та ІІ областей, тобто вони є кращими хоча б по одному із критеріїв та не гіршими по інших критеріях. Аналогічно, точки D, F, K із області ІІ не домінуються точкою М із області ІІІ. Таким чином, точки А, В, С належать до рангу 1 і складають множину оптимальних розв'язків

по Парето. Точки D, F, K належать до рангу 2. Точка M належить до рангу



3.

Рисунок 2.5 – Приклад визначення рангів відповідно до рівнів домінантності для двухкритеріальної задачі мінімізації.

На рисунках 2.6-2.8 зображено процес пошуку рішень генетичним алгоритмом у 1-му, 100-му та 200-му поколіннях.

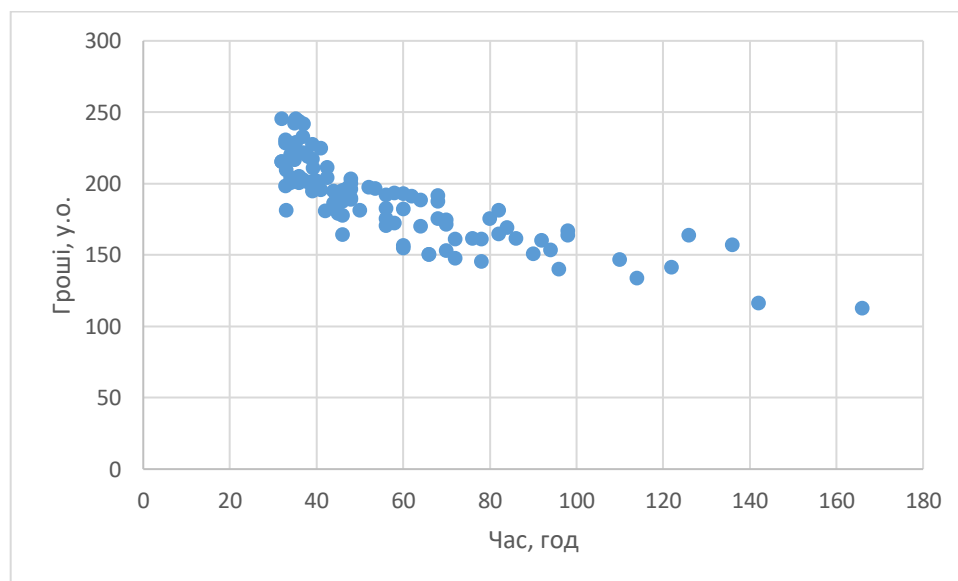


Рисунок 2.6 – Множина рішень генетичного алгоритму в 1-му поколінні

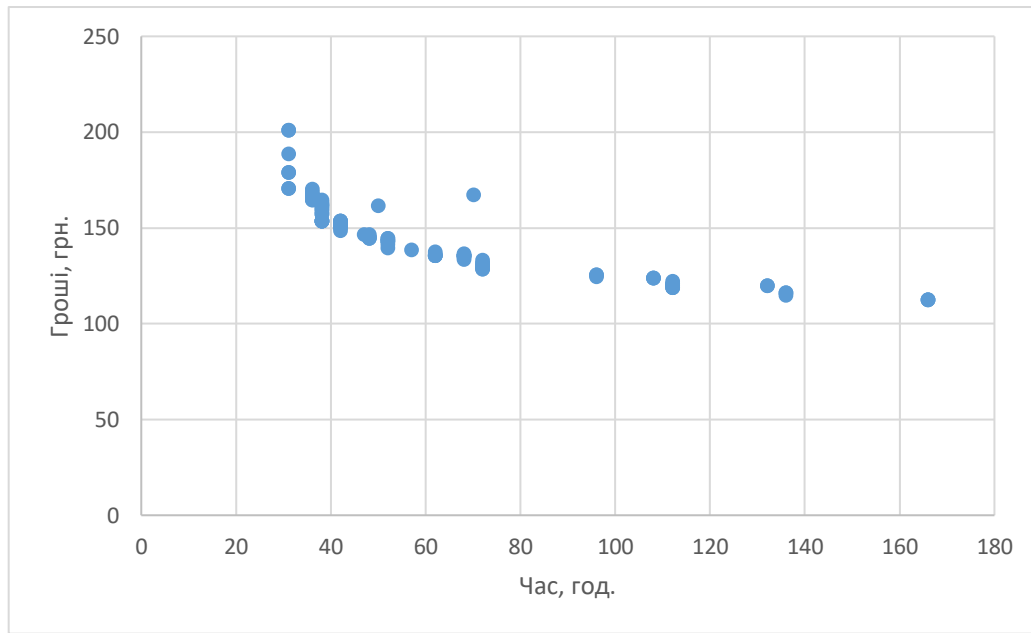


Рисунок 2.7 – Множина рішень генетичного алгоритму в 100-му поколінні

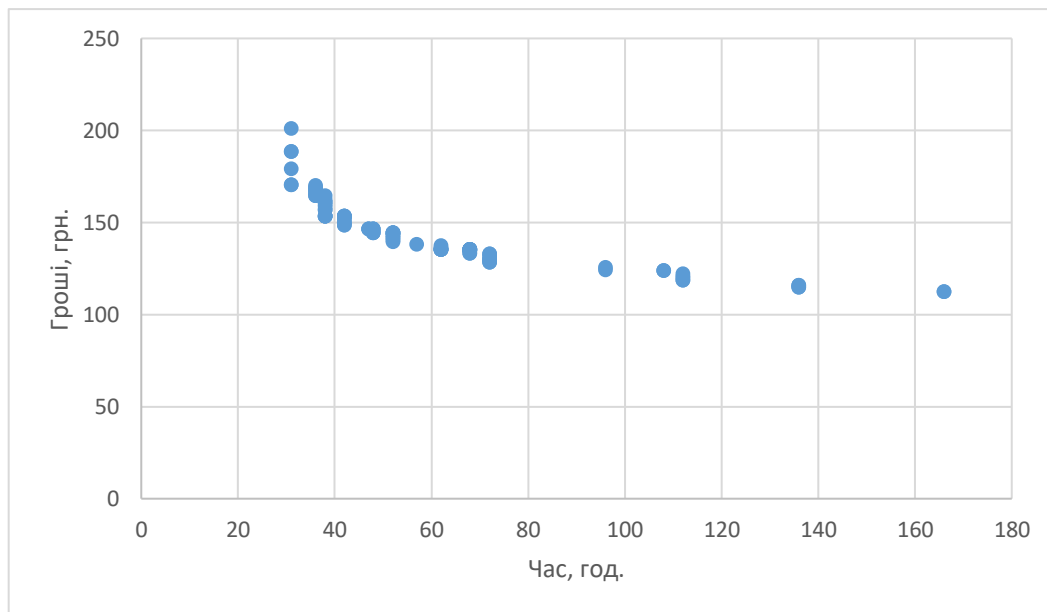


Рисунок 2.8 – Множина рішень генетичного алгоритму в 200-му поколінні

Як видно з вищенаведених рисунків, генетичний алгоритм з кожною ітерацією покращує множину кандидатів-розв'язків. Поступово розв'язки покращуються та групуються в лінію, що називається фронтом Парето та

містить множину оптимальних по Парето розв'язків. Блок-схему багатокритеріального генетичного алгоритму наведено у додатку А.

## 2.5 Експериментальні дослідження

Для розв'язання двокритеріальної задачі оптимізації за критерієм мінімізації загального часу завершення виконання робіт та критерієм мінімізації загальної суми бонусів було розроблено генетичний алгоритм для вирішення задачі багатокритеріальної оптимізації з використанням підходу Парето.

У додатку А наведено графіки зміни множини рішень задачі в процесі роботи розробленого алгоритму для 1-го, 5-го, 20-го, 50-го, 100-го, 200-го поколінь. З графіків видно, що значні покращення відбуваються вже з 20-го покоління і далі множина розв'язків змінюється незначним чином.

Розроблений алгоритм було оптимізовано за допомогою введення паралельних обчислень в програмній реалізації генетичного алгоритму. Оскільки обчислення відстані до сусідніх особин для кожної особини популяції в межах рангу є незалежними операціями, то було застосовано розпаралелювання обчислень для даного кроку алгоритму.

Для визначення ефективності розробленого алгоритму було проведено ряд експериментальних досліджень. Результати експериментів наведено в таблиці 2.1. Експерименти проводились для задач наступної розмірності: кількість виконавців  $m = 20$ , кількість робіт  $n$  змінювалася від 40 до 490 з кроком 40 робіт.

На рисунку 2.9 наведено графіки для порівняння швидкодії генетичного алгоритму та його модифікованої версії із застосуванням паралельних обчислень.

Таблиця 2.1 – Результати експериментальних досліджень часу роботи генетичного алгоритму

Кількість робіт, шт.	Час роботи генетичного алгоритму, сек	Час роботи генетичного алгоритму з використанням паралельних обчислень
40	116	102
90	303	260
140	480	430
190	575	536
240	377	326
290	461	417
340	567	542
390	709	516
440	835	642
490	1246	775

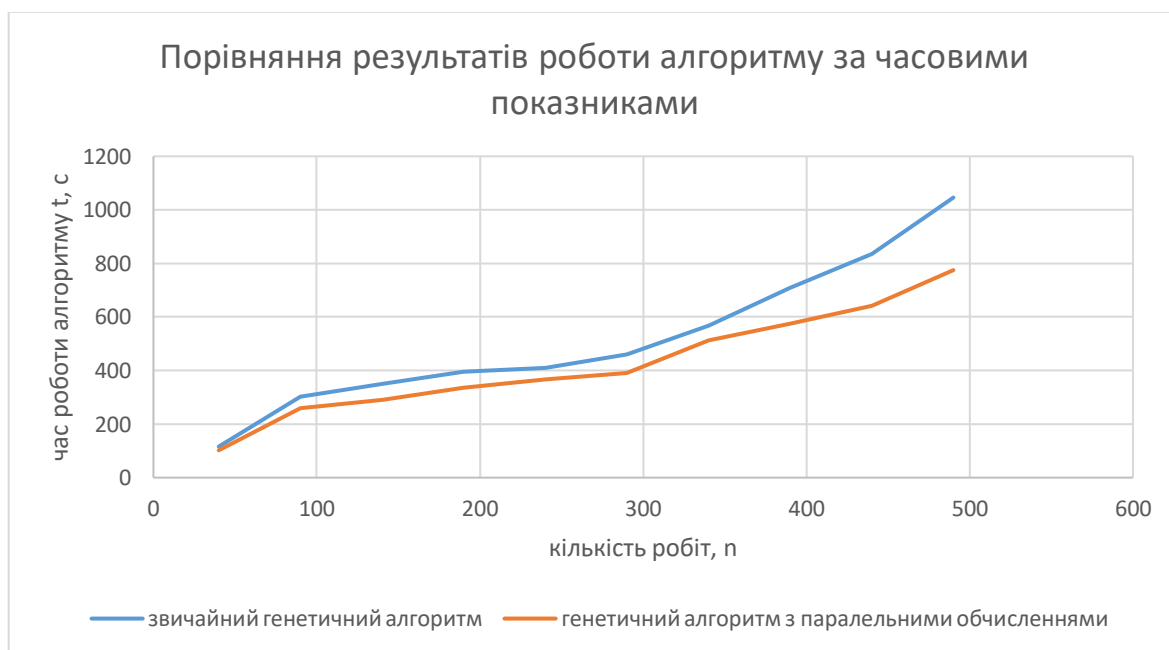


Рисунок 2.9 – Порівняння швидкодії алгоритмів

Як видно із результатів досліджень, застосування паралельних обчислень в генетичному алгоритмі дає вииграш у швидкодії майже на 16% в порівнянні із його звичайною версією.

### **Висновок до розділу**

В даному розділі було сформульовано змістовну постановку задачі та побудовано відповідну математичну модель. Було розглянуто методи розв'язання задач, що відповідають першому критерію поставленої задачі, а саме мінімізації загального часу завершення виконання робіт та визначено, що евристичні методи користуються великою популярністю.

Також було розглянуто 3 різні підходи до вирішення задач багатокритеріальної оптимізації. Для кожного підходу було детально проаналізовано його переваги та недоліки. На основі проведеного аналізу було обрано підхід Парето.

Було описано схему роботи генетичного алгоритму з використанням підходу Парето для вирішення поставленої багатокритеріальної задачі складання розкладу. Також було проведено оптимізацію розробленого алгоритму за допомогою використання паралельних обчислень. Ряд проведених досліджень показали, що це дозволило зменшити час роботи алгоритму.

## 3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1 Вимоги до програмного забезпечення

Програмне забезпечення представлено у вигляді веб-застосунку із клієнт-серверною архітектурою (рисунок 3.1). Веб-застосування може інтегруватися із іншими системами або ж виступати як окрема незалежна система.

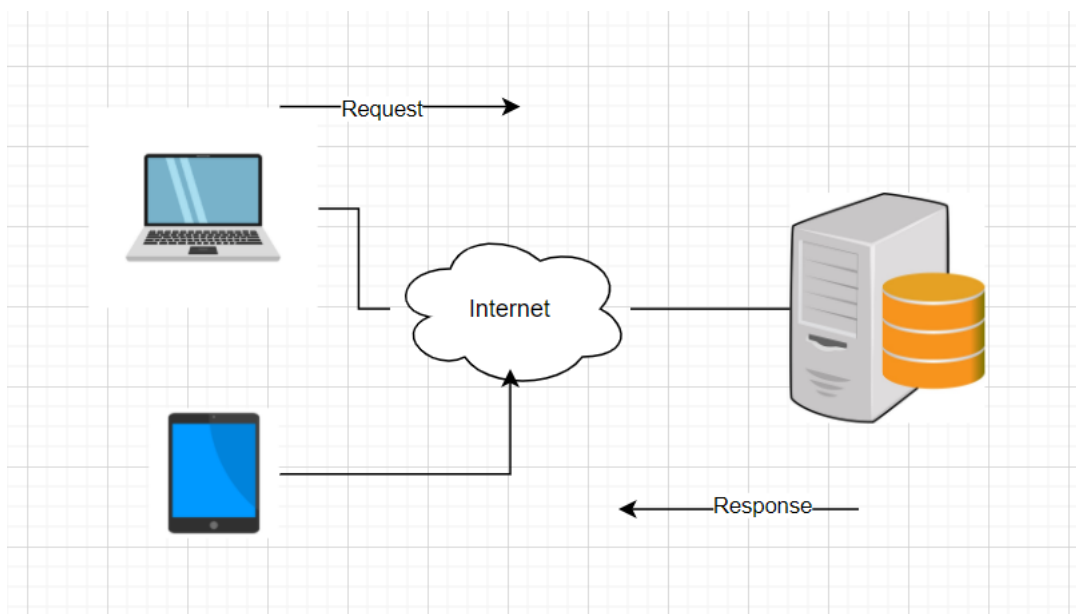


Рисунок 3.1 – Клієнт-серверна архітектура системи

Дана архітектура набула значного поширення завдяки активному розвитку мережі Інтернет та перенесення великої частини інформації на сервера та бази даних [45]. Клієнт-серверна архітектура характеризується як концепція інформаційної мережі, де велика частина ресурсів знаходиться на серверах, що обслуговують клієнтів [46].

Відповідно до схеми структурної варіантів використання, наведеної у додатку А опишемо функціональні вимоги до програмного забезпечення у таблиці 3.1.

Таблиця 3.1 – Функціональні вимоги до програмного забезпечення

№	Назва варіанту використання	Вимога	Пріоритет
1	Перегляд інформації про співробітників	Система повинна надавати можливість перегляду інформації про співробітників, зокрема, повне ім'я, посада, кваліфікація та загальна сума бонусів	Високий
2	Додавання нового співробітника	Система повинна надавати можливість додавання інформації про нового співробітника, при цьому поле з бонусами не є обов'язковим	Середній
3	Перегляд інформації про завдання	Система повинна надавати можливість перегляду списку завдань з інформацією про них, зокрема, назва, опис, час, бонуси. Після вибору варіанту розподілу кожне завдання відобразатиме свого виконавця.	Високий
4	Додавання нового завдання	Система повинна надавати можливість створення нового завдання.	середній



Продовження таблиці 3.1

№	Назва варіанту використання	Вимога	Пріоритет
5	Запуск пошуку множини оптимальних розкладів виконання завдань	Система повинна знаходити множину оптимальних розподілів завдань, на основі існуючих даних про співробітників та завдання. Після запуску алгоритму повинен з'являтися результат, у якому кожен розподіл містить інформацію про загальну суму витрачених бонусів та загальний час завершення виконання робіт	Високий
6	Вибір розкладу із множини	Система повинна надавати можливість вибору одного розкладу із множини оптимальних. Після чого кожній задачі автоматично присвоюється виконавець, а кожному виконавцю кількість бонусів.	Високий
7	Запуск візуалізації роботи алгоритму	Система повинна надавати можливість візуалізації процесу роботи алгоритму	Низький

Із таблиці 3.1 видно, що більшість вимог мають високий пріоритет, і лише незначна частина – середній та низький, що означає, що дані вимоги не мають критичного впливу на роботу системи.

### 3.2 Засоби розробки

Для розробки системи будемо використовувати наступні технології та інструменти:

- фреймворк ASP.NET Core версії 3.1 для реалізації серверної логіки системи;
- JavaScript бібліотеку ReactJs для створення графічного інтерфейсу;
- бібліотеку Net Core SignalR для передачі даних від серверу до клієнта та від клієнта до серверу в режимі реального часу;
- технологія доступу до даних Entity Framework Core;
- система управління реляційними базами даних MS SQL;
- мови програмування C# та JavaScript;
- редактор коду Visual Studio Code.

Для реалізації веб-застосунку було обрано ASP.NET Core 3.1, оскільки це найновіший фреймворк для розробки веб-застосунків на мові C#, що має ряд переваг [47].

ASP.NET Core – це кросплатформенний фреймворк із відкритим вихідним кодом. Він об'єднує в собі одночасно структуру MVC та Web API та дозволяє створювати надшвидкі API для застосунків. Фреймворк кросплатформенний, а це означає, що всі застосунки, створені на його основі, можуть працювати в різних операційних системах – Windows, Linux, Mac [48 – 50].

ASP.NET Core фреймворк забезпечує вбудовану підтримку впровадження залежностей – dependency injection. Завдяки цьому зникає необхідність встановлення таких сторонніх фреймворків як Ninject та Autofactor. Dependency injection – це свого роду шаблон, що допомагає розробнику відділяти один від одного різні частини застосунку, що робить застосування зручним для внесення змін та підтримки [51]. До виходу

ASP.NET Core єдиним рішенням для впровадження даного підходу було встановлення сторонніх вищезгаданих фреймворків. Зараз же впровадження залежностей розглядається в ASP.NET Core як об'єкт першого класу.

Для розробки графічного веб-клієнта було обрано JavaScript бібліотеку – ReactJs [52-54]. Дана бібліотека має ряд переваг:

- легка у вивченні завдяки простому синтаксису;
- високий рівень гнучкості та максимальний відклик;
- віртуальна DOM, що дозволяє впорядковувати документи HTML або XML форматів в дерево, що є найбільш підходящим для аналізу веб-браузером різних елементів застосунку;
- в поєднанні з ES6/7 легко справляється з високими навантаженнями;
- підтримує такий потік даних, при якому дочірні елементи не можуть впливати на батьківські дані;
- бібліотека з відкритим вихідним кодом, що дозволяє розробникам всього світу на основі досвіду використання покращувати та вдосконалювати фреймворк;
- простий перехід від версії до версії [55].

Entity Framework Core представляє об'єктно-орієнтовану та розширювану технологію для доступу до даних. EF Core є ORM-інструментом – відображення даних на реальні об'єкти [56]. Він надає високий рівень абстракції для роботи із даними. Таким чином, можна абстрагуватися від самої бази даних та її таблиць і працювати з даними незалежно від типу сховища. Entity Framework Core підтримує різні системи баз даних. Таким чином, можна працювати із будь-якою СУБД, якщо для неї існує відповідний провайдер.

Також варто відмітити, що EF Core надає універсальний API для роботи із даними. Тому, якщо виникне необхідність змінити цільову СУБД,

то основні зміни в проєкті будуть відноситися лише до конфігурації та налаштування підключення до відповідних провайдерів. Весь існуючий код, що стосується безпосередньо роботи із даними, залишиться без змін [57].

Бібліотека ASP.NET SignalR спрощує налаштування процесу роботи веб-функцій в режимі реального часу. Вона надає можливість миттєвого оновлення даних на клієнтах по мірі доступності сервера, не очікуючи поки клієнт надішле запит на нові дані [58].

Для збереження даних в системі використовуємо базу даних MSSQL [57]. Перевагами SQL Server є те, що СУБД здатна до масштабування. Потужний процесор здатний виконувати великий об'єм запитів одночасно. Система дозволяє обробляти транзакції в інтерактивному режимі, можна створювати профілі користувачів. Також SQL Server підтримує реплікації через інтернет та синхронізацію [60].

### **3.3 Архітектура програмного забезпечення**

Як уже згадувалося раніше, для розробки системи будемо використовувати клієнт-серверну архітектуру. Для обміну даними буде використовуватися протокол HTTP. Для миттєвого оновлення даних в реальному часі будемо застосовувати websockets. Для розробки системи було обрано монолітну багатошарову архітектуру (рисунок 3.2).

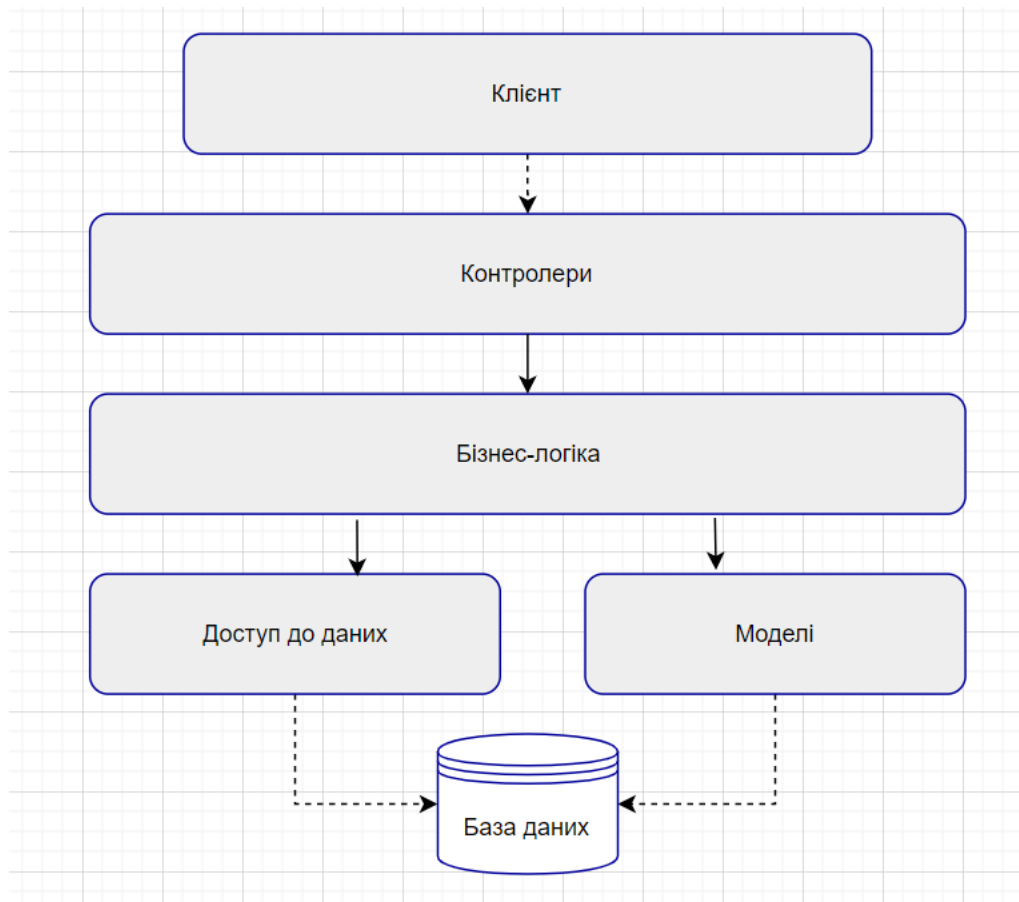


Рисунок 3.2 – Схема багатошарової архітектури програмного забезпечення

Так в системі виділяються шари користувацького інтерфейсу, бізнес-логіки та доступу до даних. В рамках такої архітектури користувачі надсилають запити через шар користувацького інтерфейсу, який взаємодіє лише із шаром бізнес-логіки. В свою чергу, шар бізнес-логіки, може використовувати шар доступу до даних для виконання запитів. Таким чином, для кожного шару виділяється окрема зона відповідальності.

В системах, що мають багатошарову архітектуру, встановлюються певні обмеження для взаємодії між шарами. Таки чином вдається підтримувати інкапсуляцію, обмежувати залежність шарів один від одного, що дозволяє зменшити наслідки внесених змін та легко змінювати функціональні можливості в рамках системи.

У ході реалізації системи було створено ряд класів. Схема структурна класів системи наведена у додатку А. Також відповідно до простору імен класів було розроблено схему пакетів системи. Схема структурна пакетів системи наведена у додатку А.

Так, розроблена архітектура дозволяє підвищити зручність підтримки системи, внесення змін та масштабування.

Для збереження даних в системі було розроблено схему база даних. Структурна схема бази даних наведена у додатку А.

Розглянемо детальніше кожен з таблиць бази даних та її поля. В таблиці Employees зберігається інформація про співробітників (таблиця 3.2). В таблиці Tasks міститься інформація про завдання (таблиця 3.3). Таблиця Schedules містить інформація про вибрані розподіли завдань(таблиця 3.4). Таблиця Bonuses зберігає інформацію про нараховані бонуси (таблиця 3.5). В таблиці EmployeesToTask міститься зв'язок між працівником та призначеною йому задачею (таблиця 3.6).

Таблиця 3.2 – Опис таблиці Employees

Поле	Тип даних	Призначення
id	int	Унікальний ідентифікатор співробітника в системі
name	string	Повне ім'я співробітника
position	string	Позиція в компанії
qualification	string	Рівень кваліфікації співробітника
speed	float	Продуктивність роботи

Таблиця 3.3 – Опис таблиці Tasks

Поле	Тип даних	Призначення
id	int	Унікальний ідентифікатор завдання в системі
title	string	Заголовок завдання
description	string	Детальний опис завдання
time	float	Еталонний час виконання завдання
baseBonus	float	Базова кількість бонусів за завдання

Таблиця 3.4 – Опис таблиці Schedules

Поле	Тип даних	Призначення
id	int	Унікальний ідентифікатор розкладу
createdAt	datetime	Дата створення розкладу
schedule	string	Серіалізований об'єкт розкладу
bonusAmount	float	Загальна кількість бонусів для розкладу
generalTime	float	Загальний час завершення виконання робіт для розкладу

Таблиця 3.5 – Опис таблиці Bonuses

Поле	Тип даних	Призначення
id	int	Унікальний ідентифікатор запису
employeeId	int	Ідентифікатор працівника, якому нараховано бонуси
bonuses	float	Кількість нарахованих бонусів

Таблиця 3.6 – Опис таблиці EmployeesToTask

Поле	Тип даних	Призначення
id	int	Унікальний ідентифікатор запису
createdAt	datetime	Дата створення запису
employeeId	string	Ідентифікатор працівника, якому призначається завдання
taskId	float	Ідентифікатор завдання, що призначається

### 3.4 Інструкція користувача

Для того, щоб розпочати роботу із системою, необхідно ввести адресу веб-системи у адресному рядку браузера. Після відкриття веб-сайту з'явиться головна сторінка системи. Форма головної сторінки наведена у додатку А.

Верхня панель містить головні пункти меню системи. Розглянемо детально кожен із них.



Пункт меню «Employees» - розділ з інформацією про співробітників. Даний розділ містить таблицю в якій відображається така загальна інформація про співробітників, як повне ім'я, посада, кваліфікація, кількість нарахованих бонусів. Дані в таблиці можна сортувати за будь-якою колонкою. Форма для перегляду інформації про співробітників наведено у додатку А.

Над таблицею із даними знаходиться кнопка «Add new employee», що дозволяє додати інформацію про нового співробітника. Після натискання на кнопку з'явиться форма для заповнення даних про працівника, що містить такі поля, як повне ім'я, позиція, рівень кваліфікації, коефіцієнт продуктивності та кількість бонусів. Поле «Bonuses» є обов'язковим для заповнення. Форма додавання інформації про нового співробітника наведено у додатку А.

Після натискання кнопки «Save» дані автоматично відобразяться у таблиці.

У розділі «Tasks» наведено перелік всіх завдань, що необхідно виконати. Кожне завдання містить назву, час виконання, базову кількість бонусів, опис та відповідального працівника. Форма перегляду списку завдань наведено у додатку А.

Над переліком завдань знаходиться кнопка «Add new task», що дозволяє додати нове завдання до переліку. Після натискання на кнопку відкриється форма для введення даних про нове завдання (рисунки 3.3).

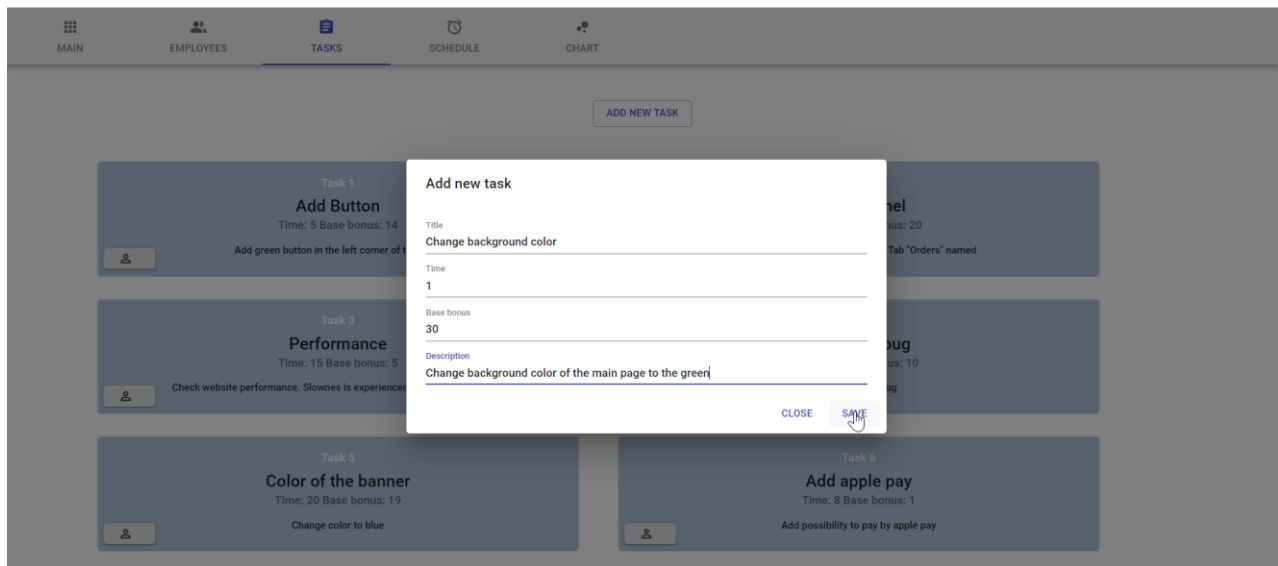


Рисунок 3.3 – Форма додавання нового завдання

Після натиснення кнопки «Save» нове завдання автоматично відобразиться в існуючому списку завдань.

Для того, щоб виконати оптимальний розподіл завдань між працівниками необхідно перейти до розділу «Schedule» та натиснути кнопку «Find schedules», що запустить алгоритм пошуку множини оптимальних розкладів відповідно до критеріїв мінімізації загального часу завершення виконання робіт та мінімізації загальних бонусних витрат (рисунок 3.4).

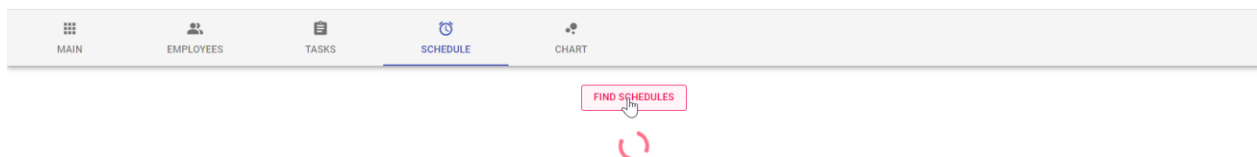


Рисунок 3.4 – Запуск алгоритму для пошуку множини оптимальних розкладів

Після завершення роботи алгоритму на сторінці з'явиться множина знайдених оптимальних рішень (рисунок 3.5).

Кожен варіант розподілу містить такі дані як загальний час завершення виконання робіт та загальна сума бонусів. Саме ці параметри відповідають критеріям пошуку рішень. Користувач має змогу оцінити

отримані результати та із врахуванням поточних пріоритетів, наприклад, час чи гроші, обрати із загальної множини результатів єдиний варіант розподілу. Для того, щоб обрати певний розклад, необхідно натиснути кнопку «Select» на відповідному варіанті.

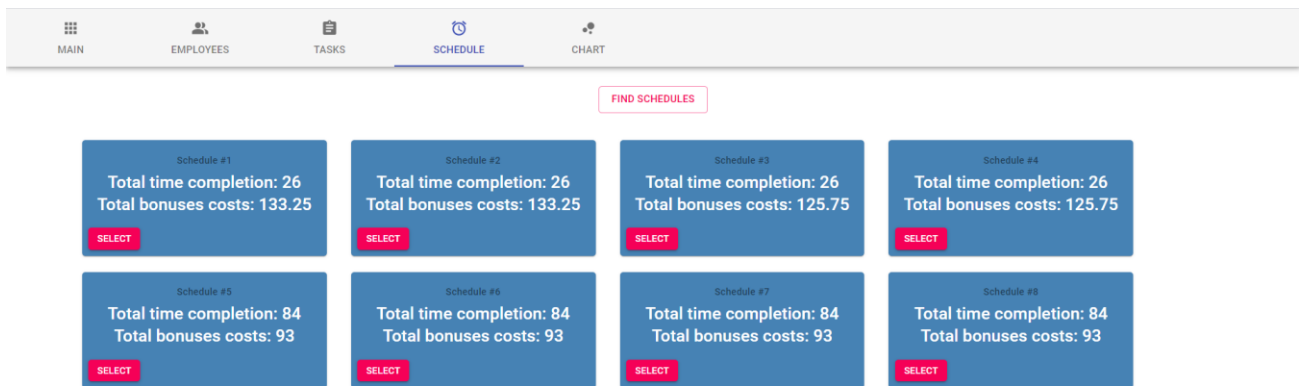


Рисунок 3.5 – Множина знайдених оптимальних розподілів завдань

Після вибору певного варіанту розподілу завдань, зміни одразу відобразяться на сторінках «Employees» та «Tasks». Зокрема, в розділі з інформацією про працівників відобразиться актуальна сума нарахованих бонусів при обраному розподілі, а на сторінці із переліком завдань, кожне завдання отримає призначеного відповідального (рисунок 3.6).

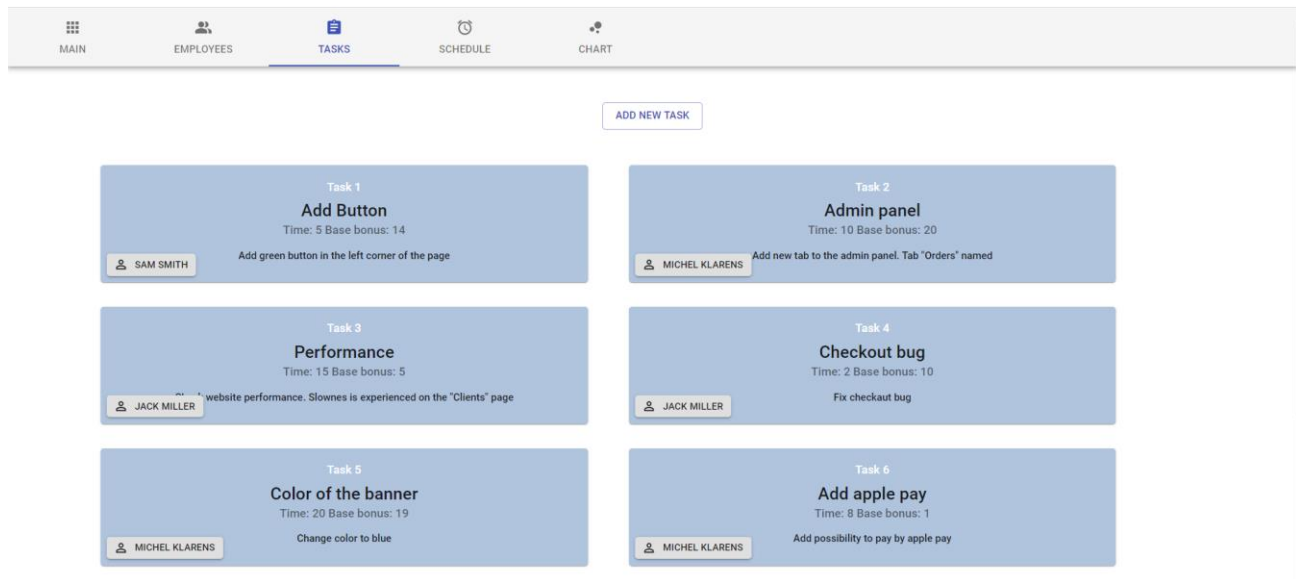


Рисунок 3.6 – Список завдань із призначеними виконавцями

У розділі «Chart» міститься графік, що демонструє процес роботи алгоритму в реальному часі. Для запуску алгоритму необхідно натиснути кнопку «Start», після чого можна спостерігати та досліджувати роботу алгоритму (рисунок 3.7).

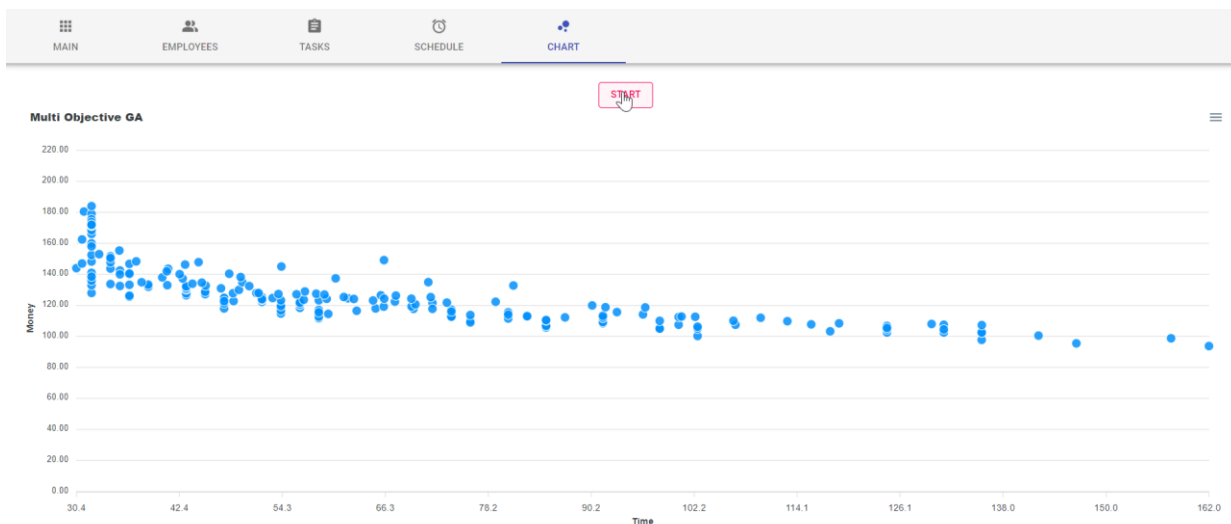


Рисунок 3.7 – Графік демонстрації процесу роботи алгоритму в реальному часі

### 3.5 Опис технічного забезпечення

Важливою складовою в процесі розробки та підтримки системи є вибір якісного технічного забезпечення, що задовольняє всі технічні вимоги до роботи системи.

Для забезпечення ефективної та стабільної роботи серверної частини програмного забезпечення необхідно забезпечити використання обладнання із наступними характеристиками:

- а) сервер з наступними характеристиками:
  - 1) тактова частота процесора не нижче 2 ГГц;
  - 2) об'єм оперативної пам'яті від 8Гб;
  - 3) SSD диск об'ємом 512Гб;
- б) Програмне забезпечення, яке необхідно встановити на сервері:
  - 1) операційна система Windows 10;
  - 2) SQL Server 2016 та вище;
  - 3) .NET Core версії 3.1 та вище;
  - 4) Node.js версії 14.15.1 та вище;
  - 5) NPM 6.14.8.

Для забезпечення правильної роботи клієнтської частини системи необхідно мати встановлений веб-браузер, наприклад, Chrom, Safari, Mozilla, та встановлений JavaScript в ньому.

#### Висновок до розділу

В даному розділі було виявлено функціональні вимоги до роботи програмного забезпечення та визначено їх пріоритети. Було обрано та описано інструменти та технології для розробки системи, зокрема серверна частина розробляється на ASP.NET Core 3.1, а клієнтська частина на ReactJs. В якості архітектури обрано монолітну багатошарову архітектуру програмного забезпечення, що спрощує внесення змін, підтримку та масштабування системи.

Для збереження даних в системі було спроектовано структурну схему бази даних, де зберігається інформація про співробітників, завдання та вибрані розклади. Також було розроблено інструкцію користувача, що детально описує можливості використання програми, та визначено вимоги до технічного забезпечення для підтримки стабільної роботи системи.

## 4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

### 4.1 Опис ідеї проекту

Пропонується створити систему для «розумного» нарахування бонусів співробітникам компанії.

Основною відмінністю даної системи є наявність алгоритмічного забезпечення для оптимізації бонусних витрат. Система не лише веде облік бонусних нарахувань співробітникам, але й генерує оптимальні розподіли завдань. При розподілі система враховує одночасно декілька критеріїв, що робить її придатною для застосування в реальних умовах компаній. Це мінімізація загальних витрат на бонусні винагороди та мінімізація загального часу завершення завдань. Внаслідок роботи алгоритму, отримуємо декілька варіантів розподілів завдань, що вважаються оптимальними відповідно до встановлених критеріїв. Таким чином, користувачу надається можливість обрати найвигідніше для себе на даний момент рішення.

Призначенням цієї системи є визначення оптимального розподілу завдань між членами команди для мінімізації загальних витрат та досягнення високої ефективності роботи.

Сутністю розробки є створення клієнт-серверної системи, веб застосунку, що легко інтегрується з іншими корпоративними системами, містить інформацію про бонусні нарахування, генерує оптимальні розподіли завдань.

Цільовою аудиторією є підприємства, в яких існує матеріальна складова мотивації працівників.

Основними вигодами використання є мінімізація витрат ресурсів компанії, зокрема грошей та часу.

## 4.2 Опис конкурентів

На момент пошуку інформації про конкурентів було виявлено 3 системи, що мають застосування в процесах бонусних нарахувань. Це SalesUp CRM, плагін Compensation for Creatio та Bonus Management System. SalesUp CRM система орієнтована лише на працівників відділів продажу, що значно звужує круг її клієнтів і не несе для нас великої загрози в якості конкурента.

Bonus Management System є автоматизованим рішенням для розрахунку та нарахування премій на базі системи управління персоналом БОС-Кадровик. Система є достатньо примітивною, дозволяє лише фіксувати нараховані суми премій, тому також не виступає для нас серйозним конкурентом.

Застосування Compensation for Creatio розроблено для компаній, які мають складну систему нарахування бонусів і винагород, має можливість налаштовування правил нарахування бонусів для співробітників та команд, реалізовано візування для отримання бонусів, наявний потужний інструмент для аналітики бонусних виплат. Нижче на рисунках 4.1-4.2 наведено приклади екранних форм даного застосунку.

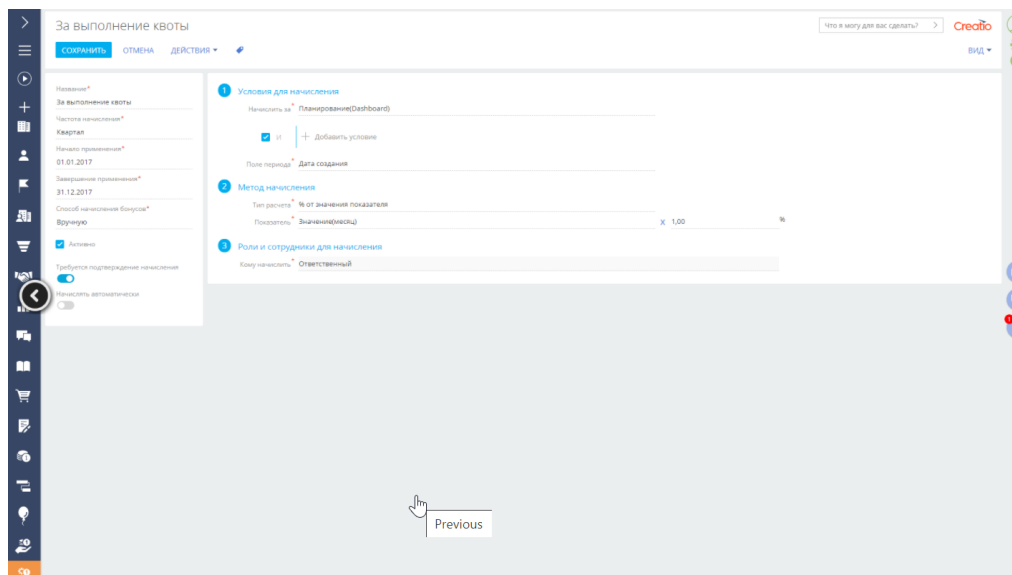


Рисунок 4.1 – Форма налаштування умов для бонусів



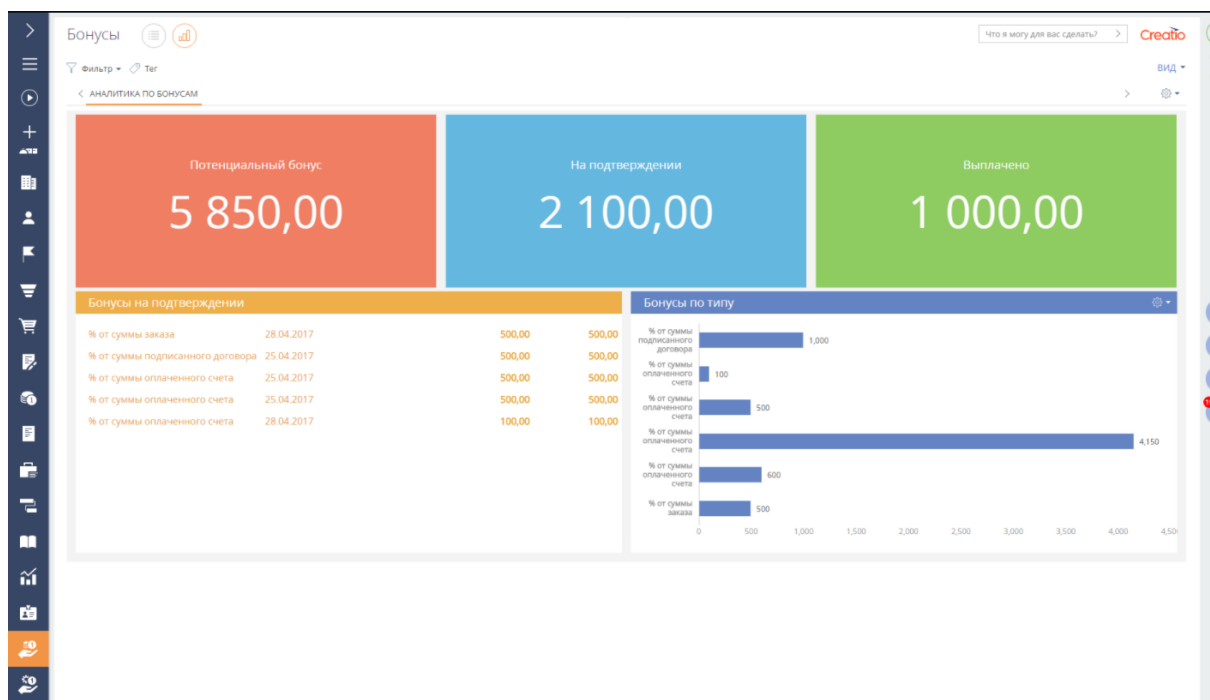


Рисунок 4.2 – Форма аналітики по бонусам

Однак даний продукт, на відміну від нашої системи, не містить алгоритмічного забезпечення для отримання кращих результатів в роботі бонусної системи підприємства.

Перед порівнянням нашого застосування та Compensation for creation виділимо основні ідеї та відобразимо їх у формі таблиць. У таблиці 4.1 наведено огляд сильних, слабких та нейтральних сторін проекту разом в порівнянні із даними конкурента.

Таблиця 4.1 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

Ідея	(потенційні) концепції		Слабка сторона (W)	Нейтральна сторона (N)	Сильна сторона (S)
	<i>BonusApp</i>	<i>Compensation for Creatio</i>			
Аналітика по бонусам	Наявний	Наявний		+	

Продовження Таблиці 4.1

Ідея	(потенційні) концепції		Слабка сторона (W)	Нейтральна сторона (N)	Сильна сторона (S)
	<i>BonusApp</i>	<i>Compensation for Creatio</i>			
Візуалізація роботи алгоритму	Наявна	Відсутня		+	
Наявність API для взаємодії з іншими системами	Наявне	Відсутнє			+
Застосування алгоритмічного забезпечення	Наявне	Відсутнє			+
Робота в реальному часі	Наявна	Наявна		+	

В результаті перевірки аналізу сильних, слабких та нейтральних сторін проекту BonusApp можна визначити, наскільки продукт є конкурентноспроможним на ринку. Отже, видно, що продукт BonusApp має сильні сторони, що дозволяють йому конкурувати з іншими системами.

### 4.3 Технологічний аудит ідеї проекту

Проаналізуємо технічні можливості для розробки проекту, оцінимо технології реалізації. У таблиці 4.2 наведено співставлення ідей проекту до технологій реалізації.

Таблиця 4.2 – Технологічна здійсненність ідеї проекту

Ідея проекту	Технології реалізації	Наявність технології	Доступність технології
Аналітика по бонусам	СУБД MSSQL для збереження даних	Наявна	Доступна
Візуалізація роботи алгоритму	SignalR	Наявна	Доступна
Наявність API для взаємодії з іншими системами	REST API	Наявна	Доступна
Застосування алгоритмічного забезпечення	.Net Core	Наявна	Доступна
Робота в реальному часі	Обмін даними через мережу Інтернет	Наявна	Доступна

Наведемо список обраних наявних технологій для реалізації проекту та їх обґрунтування:

- аналітика по бонусам – СУБД MSSQL для збереження даних про працівників, задачі та бонуси;
- візуалізація роботи алгоритму – SignalR для передачі даних в реальному часі;
- наявність API для взаємодії з іншими системами – технологія REST API;

- застосування алгоритмічного забезпечення - .NET Core для розробки алгоритмів;
- робота в реальному часі – обмін даними через мережу Інтернет.

#### 4.4 Аналіз ринкових можливостей запуску стартап-проекту

Перед запуском стартап-проекту необхідно спочатку проаналізувати його ринкові можливості. Оцінимо основних гравців на ринку для проведення аналізу на обсяг попиту. Результати наведено у таблиці 4.3

Таблиця 4.3 – Попередня характеристика потенційного ринку стартап-проекту

Показник стану ринку	Характеристика
Кількість головних гравців	Compensation for Creatio
Загальний обсяг продаж	Клієнти в більш ніж 100 країнах світу, майже 100 тисяч користувачів системи Creatio, для якої розроблено дане розширення для бонусів
Динаміка ринку	Спостерігається постійний ріст оборотів компанії
Наявність обмежень для входу	Створення продукту та стартовий капітал
Специфічні вимоги до стандартизації та сертифікації	Немає
Середня норма рентабельності в галузі	Невідома

У таблиці 4.4 наведено опис цільових груп проекту.

Таблиця 4.4 – Характеристика потенційних клієнтів стартап-проекту

<b>Потреба</b>	<b>Цільова аудиторія</b>	<b>Відмінності у поведінці цільових груп</b>	<b>Вимоги споживачів до товару / послуги</b>
Мінімізація витрат ресурсів компанії, матеріальна мотивація співробітників	Компанії, що мають матеріальну складову мотивації	Відсутні	Зручний інтерфейс та можливість інтеграції із зовнішніми системами
			Автоматизація розподілу завдань в команді
			Облік бонусних витрат

Далі проаналізуємо ринкове середовище, визначимо фактори загроз та можливостей.

Найпоширенішою тактикою боротьби із загрозами є визначення можливих джерел виникнення загроз та створення плану реагування на їх, обробку та мінімізацію руйнівних наслідків. Наведемо існуючі та можливі майбутні загрози та опишемо шляхи їх вирішення у таблиці 4.5.

Таблиця 4.5 – Фактори загроз

<b>Фактор</b>	<b>Зміст загрози</b>	<b>Реакція компанії</b>
Кадри	Загроза некомпетентності кадрів в команді.	Удосконалення менеджменту в компанії, проведення стажувань та навчань для залучення талановитих спеціалістів

Продовження Таблиці 4.5

Фактор	Зміст загрози	Реакція компанії
Мала кількість клієнтів	Продукт має малу кількість клієнтів	Аналіз маркетингової компанії та її покращення, пошук, партнерів, проведення опитування цільової аудиторії та удосконалення продукту, створення вигідних умов для існуючих клієнтів
Недостатня кількість інвестицій	Компанія отримує недостатню кількість інвестицій для розвитку продукту	Оптимізація загальних витрат, пошук інвесторів

Розглянемо фактори можливості, що можуть мати позитивний вплив на розвиток проекту. У таблиці 4.6 наведено детальний опис.

Таблиця 4.6 – Фактори можливостей

Фактор	Зміст можливості	Реакція компанії
Науково-технічний прогрес	Виникнення нових технологій, підходів, апаратного забезпечення, що дозволяє спростити розробку та підтримку системи, зменшити витрати, покращити продукт	Підвищення кваліфікації працівників, впровадження нових технологій

Продовження Таблиці 4.6

Фактор	Зміст можливості	Реакція компанії
Світове визнання	Отримання нагород від відповідних експертів у сфері програмного забезпечення, входження в топ систем для даної предметної області	Постійна робота над розвитком та вдосконаленням системи, аналіз конкурентів
Вихід на іноземний ринок	Залучення іноземних клієнтів	Впровадження локалізація, робота над інтернаціоналізацією та сертифікацією продукту
Партнерство	Співпраця із компаніями для дослідження роботи бонусних систем	Укладання договорів із партнерами для наукових досліджень роботи бонусних систем, інвестиції в проведення наукових робіт в сфері мотивування співробітників

На основі проведеного аналізу можна зробити висновки, що мала кількість клієнтів та недостатнє фінансування є найбільшими можливими загрозами на початку запуску стартапу. Саме тому для початку необхідно зосередитися на пошуку інвесторів, пошуку клієнтів та їх утримання, маркетингу.

У таблиці 4.7 наведено аналіз існуючих пропозицій на ринку та виділено загальні риси характеристики.

Таблиця 4.7 – Ступеневий аналіз конкуренції на ринку

<b>Особливості конкурентного середовища</b>	<b>В чому проявляється дана характеристика</b>	<b>Вплив на діяльність підприємства</b>
Монополія	В галузі домінує одна фірма	Надання конкурентоспроможних послуг
Національний рівень конкурентної боротьби	Конкуренція фірм на рівні держави	Реклама
Внутрішньогалузева ознака	В галузі домінує декілька конкуруючих фірм	Конкурентоспроможні послуги
Національний рівень конкурентної боротьби	Постачальники послуг працюють в межах однієї галузі	Надання схожих за видом послуг
Товарно-видова конкуренція	Конкуренція між послугами одного виду та призначення	Вдосконалення послуг



Продовження Таблиці 4.7

<b>Особливості конкурентного середовища</b>	<b>В чому проявляється дана характеристика</b>	<b>Вплив на діяльність підприємства</b>
Нецінові переваги	Переваги відносно результату послуг	Оптимізація витрат та підвищення лояльності до замовника послуг
Марочна інтенсивність	Вказує, яке підприємство відповідальне за готовий продукт	Створення власної марки

На основі проведеного аналізу ринку в якості початкової стратегії можна обрати пошук нових та утримання існуючих клієнтів.

У таблиці 4.8 наведено аналіз умов конкуренції у галузі.

Таблиця 4.8 – Аналіз конкуренції в галузі за М. Портером

	<b>Конкуренти</b>		<b>Постачальники</b>	<b>Клієнти</b>	<b>Товари замітники</b>
	<b>Прямі</b>	<b>Потенційні</b>			
<i>Складові аналізу</i>	<i>Compensation for Creatio</i>	Невідомі	Значення розміру поставок	Розмір закупівель послуг	Загроз не існує
<i>Висновки</i>	Існує	Дані відсутні	Умови не диктуються постачальниками послуг	Встановлення вимог до вартості	Відсутні обмеження

У таблиці 4.9 наведено фактори конкурентноспроможності.

Таблиця 4.9 – Обґрунтування факторів конкурентоспроможності

<b>Фактор конкурентоспроможності</b>	<b>Обґрунтування</b>
Точність	Вдосконалення алгоритмів для покращення результатів, впровадження нових бонусних стратегій
Додатковий функціонал	Можливість налаштування нових бонусних стратегій, створення особистих кабінетів

У таблиці 4.10 описано сильні та слабкі сторони стартапу, проведено їх аналіз та визначено результати.

Таблиця 4.10 – Порівняльний аналіз сильних та слабких сторін проекту

<b>Фактор конкурентоспроможності</b>	<b>Бали 1-20</b>	<b>Рейтинг товарів конкурентів у порівнянні з КСК</b>						
Точність	17							
Додатковий функціонал	18							

Підсумуємо всю інформацію щодо аналізу ринкових можливостей у матриці SWOT-аналізу (таблиця 4.11).

Таблиця 4.11 – SWOT- аналіз стартап-проекту

<b>Сильні сторони</b>	<b>Слабкі сторони</b>
наявність API для інтеграції з іншими системами аналітика по бонусам візуалізація роботи алгоритму робота в реальному часу застосування алгоритмічного забезпечення	потрібні значні інвестиції для розробки та запуску проекту
<b>Можливості</b>	<b>Загрози</b>
науково-технічний прогрес світове визнання вихід на іноземний ринок партнерство	кадри мала кількість клієнтів недостатня кількість інвестицій

Враховуючи сильні сторони та можливості проекту, наведені у матриці SWOT-аналізу, можна зробити висновок, що продукт має право на життя та, крім того, є конкурентноспроможним.

#### 4.5 Розроблення ринкової стратегії проекту

Для того, щоб розробити ринкову стратегію, необхідно визначити охоплення ринку за допомогою визначення груп потенційних споживачів (таблиця 4.12).

Таблиця 4.12 – Вибір цільових груп потенційних споживачів

<b>Опис профілю цільової групи</b>	<b>Готовність споживачі в сприйняти послугу</b>	<b>Орієнтовний попит в межах цільової групи</b>	<b>Інтенсивність конкуренції в сегменті</b>	<b>Простота входу в сегмент</b>
ІТ-компанії	Готові	Високий	Невідома	Висока
Інші компанії	Середній	Середній	Невідома	Низька

В якості цільових груп було обрано ІТ-компанії та інші компанії, які мають матеріальну складову мотивації персоналу. Простота входу в сегмент для першої цільової групи є високою, оскільки ІТ-компанії легко інтегрують нові автоматизовані рішення для спрощення роботи.

Базова стратегія розвитку наведена в таблиці 4.13

Таблиця 4.13 – Визначення базової стратегії розвитку

<b>Обрана альтернатива розвитку проекту</b>	<b>Стратегія охоплення ринку</b>	<b>Ключові конкурентоспроможні позиції відповідно до обраної альтернативи</b>	<b>Базова стратегія розвитку</b>
Стратегія спеціалізації	Пошук контактів потенційних клієнтів, утримання існуючих клієнтів, можливість налаштування системи відповідно до вимог клієнта та надання допомоги з інтеграцією.	Створення якісного продукту і його постійне вдосконалення, розробка системи лояльності для клієнтів, орієнтація на довготривалу співпрацю.	Стратегія диференціації

В якості базової стратегії було обрано стратегію диференціації, що означає орієнтація на потреби користувача. В якості альтернативної стратегії вибрано стратегію спеціалізації, що означає підлаштування на окремий цільовий сегмент.

Визначимо стратегію конкурентної поведінки на ринку (таблиця 4.14).

Таблиця 4.14 – Визначення базової стратегії конкурентної поведінки

<b>Чи є проект «першопрохідцем» на ринку?</b>	<b>Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?</b>	<b>Чи буде компанія копіювати основні характеристики товару конкурента, і які?</b>	<b>Стратегія конкурентної поведінки</b>
Ні	Компанія буде шукати нових споживачів та намагатися залучати існуючих клієнтів конкурентів	Компанія не буде копіювати основні характеристики продукту конкурента	Стратегія заняття конкурентної ніші

Наведемо результуючий список визначених стратегій:

- базова стратегія розвитку – диференціація;
- альтернативна стратегія розвитку – спеціалізація;
- конкурентна поведінка – заняття конкурентних ніш;

– сегменти ринку – ІТ-компанії та інші компанії.

#### 4.6 Розроблення маркетингової програми стартап-проекту

Опишемо маркетингову концепцію продукту, який буде пропонуватися кінцевому користувачу.

Для цього у таблиці 4.15 приведемо результати аналізу конкурентноспроможності продукту, що проводився раніше.

Таблиця 4.15 – Визначення ключових переваг концепції потенційного товару

Потреба	Вигода, яку пропонує товар (послуга)	Ключові переваги перед конкурентами
Зручність нарахування бонусів співробітникам, підвищення ефективності роботи працівників при мінімальних витратах	Система знаходить компромісні розподіли завдань в команді, при якому досягається висока продуктивність при мінімальних витратах. Кінцевий користувач має можливість обрати вигідний для себе варіант рішення із запропонованої системою множини рішень	Наявність алгоритмічного забезпечення, оптимальний розподіл завдань в команді, мінімізація бонусних витрат

Отже, можна зробити висновок, що продукт відповідає потребам користувачів та має переваги над конкурентами.

Для того, щоб уточнити ідею продукту та його особливості, побудуємо трирівневу маркетингову модель продукту (таблиця 4.16).

Таблиця 4.16 – Опис трьох рівнів моделі товару

Рівні товарів	Сутність та складові		
I. Товар за задумом	Система, що дозволяє підвищити ефективність роботи команди та оптимізувати матеріальну складову мотиваційної системи.		
II. Товар у Реальному виконанні	Властивості/характеристики	$M/$ $H_m$	$Vp/Tx$ $/Tл/E/$ $Op$
	Якість	Hм	Tл
	Точність	Hм	Tл
	Ціна	Hм	E
	Якість: тестування системи відповідно до сценаріїв		
	Пакування: веб-застосування, що відкривається за допомогою веб-браузера, має адаптовану мобільну версію.		
Марка: Найменування продукту			
III. Товар із підкріпленням	Програма лояльності для клієнтів, надання безкоштовної допомоги при інтегруванні системи.		
За рахунок чого потенційний товар буде захищено від копіювання: реєстрація патенту на винахід, захист інтелектуальної власності			

Далі необхідно визначити межі цін на продукт, враховуючи попит на продукт, витрати на розробку та підтримку системи. Ми не можемо оцінити вартість розробки та підтримки роботи програмного забезпечення конкурентів.

У таблиці 4.17 наведено межі встановлення цін. Оскільки продукт має алгоритмічне забезпечення, що здатне допомагати компанії ефективно управляти своїми ресурсами та зменшувати витрати, що є великою перевагою над конкурентами, то не варто встановлювати ціни нижче, ніж в конкурентів. Також врахуємо стереотип, що низька ціна одразу викликає підозру про неякісність послуги або продукту. Таким чином для продажу продукту будемо використовувати схему підписки. Для команд з кількістю до 10 користувачів продукт буде безкоштовним. Для всіх інших 200 грн. за користувача в місяць. Обираючи ціну орієнтувалися на ціни схожих продуктів.

Таблиця 4.17 – Визначення меж встановлення ціни

<b>Рівень цін на товари-замінники</b>	<b>Рівень цін на товари-аналоги</b>	<b>Рівень доходів цільової групи споживачів</b>	<b>Верхня та нижня межі встановлення ціни на товар/послугу</b>
Відсутні товари замінники	175 – 350 грн. за користувача в місяць	Високий	175-200 грн. за користувача в місяць, безкоштовно для команд до 10-ти чоловік

Визначимо систему збуту відповідно до наступних вимог (таблиця 4.18).



Таблиця 4.18 – Формування системи збуту

<b>Специфіка закупівельної поведінки цільових клієнтів</b>	<b>Функції збуту, які має виконувати постачальник товару</b>	<b>Глибина каналу збуту</b>	<b>Оптимальна система збуту</b>
Покупка підписки та інтегація продукту з системою користувача або ж доступ до веб-системи онлайн	Постачальник виконує такі основні функції: – розробка; – вдосконалення; – зберігання; – збут.	Канал першого рівня	Пряма система збуту з пошуком клієнтів

Розробимо концепцію маркетингової комунікації, за основу якої візьмемо стратегії позиціювання та специфіку поведінки клієнта (таблиця 4.19).

Таблиця 4.19 – Концепція маркетингових комунікацій

<b>Специфіка поведінки цільових клієнтів</b>	<b>Канали комунікації</b>	<b>Ключові позиції для позиціювання</b>	<b>Концепція рекламного звернення</b>
Впровадження бонусних систем, мінімізація витрат	Інтернет ресурси, конференції	Канал першого рівня	Підвищення ефективності роботи співробітників та мінімізація бонусних витрат завдяки впровадженню нашої системи

Як результат досліджень у даному розділі було виділено основні переваги над товарами та послугами конкурентів. Виявили канали збуту та комунікації із потенційними клієнтами. Визначили цінові межі продукту.

### **Висновок до розділу**

В даному розділі було проведено розробку стартап проекту. Було сформульовано ідею проекту, яка робить його потрібним на ринку. Також було виконано пошук та аналіз конкурентів. З отриманих результатів визначили, що система має значні переваги над конкурентами та є конкурентноспроможною на ринку.

Далі визначили технології для розробки проекту та визначили наявність та доступність кожної з них. Так, всі необхідні для розробки технології доступні до застосування. В результаті проведення аналізу ринкової стратегії визначили, що цільовою аудиторією виступають компанії, що мають матеріальну складову мотиваційної системи. Визначили фактори ризиків та сценарії реагування на їх, а також фактори можливостей для росту та розвитку проекту.

Останнім кроком було визначення маркетингової стратегії виходу стартапу на ринок. Для цього визначили вигоду, яку отримує клієнт від продукту, канали комунікації із клієнтами, канали збуту та ключові переваги над конкурентами. Основною перевагою виступає алгоритмічне забезпечення системи. Також було визначено цінові границі для продажу, при цьому невеликі команди отримають можливість користуватися системою безкоштовно.

## ВИСНОВКИ

Дана магістерська робота присвячена актуальній проблемі мотивації співробітників. Відомо, що мотиваційна система є важливою складовою організації високоефективної роботи в компанії, оскільки вона має значний вплив на продуктивність та якість роботи співробітників. Правильно обрана система мотивації дозволяє досягти високої ефективності роботи при оптимальних витратах ресурсів компанії.

Для вирішення поставленої задачі було проаналізовано ряд наукових робіт, присвячених темі мотивації працівників. Також детально досліджено предметну область та описано бізнес-процеси, що відбуваються при застосуванні матеріальної складової мотиваційної програми. Було виявлено декілька аналогів систем нарахування бонусів та визначено недоліки кожної з них.

Для забезпечення високої ефективності роботи при оптимальних витратах ресурсів, тобто, зменшення часу завершення виконання робіт при мінімальних витратах бюджету, було розроблено математичну модель. На основі математичної моделі визначили методи для розв'язання поставленої задачі. Було реалізовано багатокритеріальний генетичний алгоритм для вирішення задачі багатокритеріальної оптимізації. Також, виконали оптимізацію розробленого алгоритму для зменшення часу його виконання. Завдяки цьому швидкодію розробленого алгоритму було підвищено на 16% в порівнянні із звичайною версією алгоритму.

Для реалізації інформаційної системи було спроектовано архітектуру програмного забезпечення та обрано найновіші, найпопулярніші технології та інструменти розробки. Розроблена інформаційна система була детально протестована для забезпечення високої якості продукту.

## СПИСОК ПОСИЛАНЬ

1. Казмірчук А.В. «Інформаційна система нарахування бонусів співробітникам компанії»/ М.О. Сперкач // Матеріали VI всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2020) – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського», 26-27 листопада 2020 р. – С.44-47.
2. Казмірчук А.В. «Моделювання стратегій нарахування бонусів співробітникам компанії»/ М.О. Сперкач, О.Г. Жданова// Матеріали XV міжнародної науково–практичної конференції «Математичне та імітаційне моделювання систем МОДС 2020», – м. Чернігів, 29 червня – 1 липня, 2020 р. – С. 352-355.
3. Казмірчук А.В. «Розв’язання багатокритеріальної задачі складання розкладу з використанням генетичного алгоритму»/ М.О. Сперкач, О.Г. Жданова// Науково-технічний журнал «Сучасні інформаційні системи» - 2021.
4. Перевозчикова Н. О. Особливості мотивації персоналу на сучасних підприємствах [Електронний ресурс] / Н. О. Перевозчикова, Ю. І. Котова // Ефективна економіка. – 2013. – Режим доступу до ресурсу: <http://www.economy.nayka.com.ua/?op=1&z=1961>.
5. Гайдученко С. О. Тексти лекцій з навчальної дисципліни "Мотивація персоналу" / С. О. Гайдученко. – Харків, 2013. – 111 с.
6. Тростянецька Е. В. «Удосконалення мотивації праці в системі управління підприємством» / Е. В. Тростянецька, Г. Ю. Гупалюк. // Розвиток методів управління та господарювання на транспорті. – 2016. – №4.
7. Урманов Ф. Ш. «Мотивація – основний чинник ефективності управлінської праці» / Ф. Ш. Урманов. // ВІСНИК ЖДТУ. – 2017. – №4.

8. Устіловська А. С. «Мотивація персоналу як один з основних інструментів успішного управління персоналом» / А. С. Устіловська. // Молодий вчений. – 2017. – №4.
9. Шкода Т. Н. Матеріальне стимулювання персоналу / Т. Н. Шкода., 2011. – (Науковий вісник Полтавського університету економіки та торгівлі).
10. Чебан А. А. Ефективна система мотивації праці як елемент підвищення конкурентноспроможності підприємства / А. А. Чебан., 2015. – (Молодий вчений).
11. 6 основных KPI для оценки службы поддержки специалистов Help Desk [Електронний ресурс] // Okdesk. – 2016. – Режим доступу до ресурсу: <https://okdesk.ru/blog/helpdesk-kpi>.
12. Сложная техническая поддержка и мотивация [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://vc.ru/life/55355-slozhnaya-tehnicheskaya-podderzhka-i-motivaciya-chast-pervaya>.
13. SalesUp [Електронний ресурс] – Режим доступу до ресурсу: <https://salesap.ru/help/info/kak-nachislyat-sotrudnikam-zarplatu-i-nbsp-bonusy/>.
14. Compensation for Creatio [Електронний ресурс] – Режим доступу до ресурсу: [https://marketplace.terrasoft.ru/app/compensation-creatio?\\_ga=2.253395721.1243142852.1603629511-949484064.1589040426](https://marketplace.terrasoft.ru/app/compensation-creatio?_ga=2.253395721.1243142852.1603629511-949484064.1589040426).
15. БОСС. Кадровые системы [Електронний ресурс] – Режим доступу до ресурсу: <https://boss.ru/solutions/bms/>.
16. Pinedo M. L. Scheduling. Theory, Algorithms, and Systems / Michael Pinedo., 2012.
17. Зак Ю.А. Прикладные задачи теории расписаний и маршрутизации перевозок / Ю.А. Зак. – М.: Книжный дом «ЛИБРОКОМ», 2012. – 394 с.
18. Пойа Д. Как решать задачу / Д. Пойа., 1959. – 208 с.
19. Клейнберг Д. Алгоритмы. Разработка и применение / Д. Клейнберг, Е. Тардос., 2016. – 800 с.

20. Адамар Ж. Исследование психологии процесса изобретения в области математики. М., "Сов. радио", 1970.
21. Рафгарден Т. Совершенный алгоритм. Жадные алгоритмы и динамическое программирование / Тим Рафгарден., 2012.
22. Жадные алгоритмы [Электронный ресурс] – Режим доступа до ресурсу: <http://www.williamspublishing.com/PDF/5-8459-0857-4/part.pdf>.
23. Deb K. Understanding Interactions Among Genetic Algorithm Parameters / K. Deb, S. Agrawal. – 1998.
24. Гладков Л.А. Генетические алгоритмы / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик ; под ред. В.М. Курейчика. – 2-е изд., испр. и доп. – М.: ФИЗМАТЛИТ, 2006. – 320 с.
25. Еремеев А.В. Генетический алгоритм для задачи о покрытии / А.В. Еремеев // Дискрет. анализ и исслед. операций. Сер. 2. – 2000. – Т. 7, № 1. – С. 47-60.
26. Аверченков В. I. Эволюционное программирование и его применение / В. I. Аверченков., 2016.
27. Гладков Л. Генетичні алгоритми / Л. Гладков., 2010.
28. Калашников А.В. Параллельный алгоритм имитации отжига для построения многопроцессорных расписаний / А.В. Калашников, В.А. Костенко // Известия РАН. Теория и системы управления. – 2008. – № 3. – С. 101-110.
29. Land A.H. An automatic method of solving discrete programming problems / A.H. Land, A.G. Doig // Econometrica. – 1960. – V. 28. – P. 497-520.
30. S. Vairam, V. Selladurai, Parallel machine shop scheduling using memetic algorithm. Applied Mechanics & Materials, 2014.
31. Vairam S., Selladurai V. Permutation genetic algorithm based encoding method for parallel machine scheduling and balancing. Applied Mechanics & Materials, 2014.

32. Senthil Kumar, Selladarui V., Raja K., Eleganova K. Ant Colony Approach for Makespan Minimization on Unrelated Parallel Machines. International Journal of Engineering Science & Technology, 2011.
33. Helal M. A Tabu Search Algorithm to Minimize the Makespan for the Unrelated Parallel Machines Scheduling Problem with Setup Times / Magdy Helal., 2006. – (International Journal of Operations Research).
34. German Y. Scheduling to Minimize Makespan on Identical Parallel Machines / Yousef German., 2016. – (International Journal of Scientific & Engineering Research).
35. Zhenmin Cheng. An approximate algorithm for parallel machine scheduling problem to minimize total completion time. Computing & Information Systems, 2010.
36. Багатокритеріальна оптимізація [Електронний ресурс] – Режим доступу до ресурсу: [http://eos.ibi.spb.ru/umk/4\\_4/5/print/5\\_R1\\_T8.pdf](http://eos.ibi.spb.ru/umk/4_4/5/print/5_R1_T8.pdf).
37. Штойер Р. Многокритериальная оптимизация / Р. Штойер., 1992. – 504 с.
38. Попов В. П. Интервальный подход к оптимизации решения многокритериальной задачи о назначениях / В. П. Попов., 2015.
39. Carlos, A. Coello Coello (2018). Multi-objective Optimization. Springer International Publishing [Springer International Publishing], available at: [https://link.springer.com/referenceworkentry/10.1007%2F978-3-319-07153-4\\_17-1](https://link.springer.com/referenceworkentry/10.1007%2F978-3-319-07153-4_17-1) (last accessed September 18, 2020).
40. Chiandussi, G., Codegone, M. (2012). Comparison of multi-objective optimization methodologies for engineering applications. Science Direct [Science Direct], available at: <https://www.sciencedirect.com/science/article/pii/S0898122111010406> (last accessed September 22, 2020).
41. Полковникова, Н.А., Курейчик, В.М. Многокритериальная оптимизация на основе эволюционных алгоритмов. Izvestiya SFedU.

Engineering Sciences [Izvestiya SFedU. Engineering Sciences] (останнє звернення 01.10.2020).

42. Ногин В. Д. Сужение множества Парето. Аксиоматический подход / В. Д. Ногин., 216. – 272 с.

43. Ногин В. Д. Принятие решений в многокритериальной среде / В. Д. Ногин. – Москва: Физматлит, 2002. – 144 с.

44. Ashis K. M. Multi-Objective Genetic Algorithm / K. M. Ashis, K. M. Anil. // Research Gate. – 2012.

45. Фаулер М. Архитектура корпоративных программных приложений / Мартин Фаулер., 2006. – 544 с.

46. Змерзлий І. Клієнт-серверна архітектура та ролі серверів [Електронний ресурс] / Іван Змерзлий // Medium. – 2017. – Режим доступу до ресурсу: <https://medium.com/@IvanZmerzlyi>

47. Фримен А. ASP.NET Core MVC с примерами на C# для профессионалов / Адам Фримен., 2016. – 992 с.

48. Чамберс Д. ASP.NET Core. Разработка приложений / Д. Чамберс, Д. Пэккетт., 2018. – 464 с.

49. ASP.NET documentation [Електронний ресурс] // Microsoft – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-5.0>.

50. Korotjuk J. Почему компании переходят на ASP.NET Core для разработки веб-приложений для решения своих бизнес задач [Електронний ресурс] / Julija Korotjuk. – 2020. – Режим доступу до ресурсу: <https://tqm.com.ua/likbez/article/pochemu-net-ru>.

51. Seemann M. Dependency Injection in .NET / Mark Seemann., 2011. – 520 с.

52. React vs Angular vs Vue.js — What to choose in 2020? [Електронний ресурс] // TechMagic. – 2018. – Режим доступу до ресурсу:



<https://medium.com/techmagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d>.

53. Порселло Е. React и Redux: функциональная веб-разработка / Е. Порселло, А. Бэнкс., 2018. – 336 с.

54. Тиленс М. Т. React в действии / Марк Томас Тиленс., 2019. – 368 с.

55. Mardan A. React Quickly: Painless web apps with React, JSX, Redux, and GraphQL / Azat Mardan..

56. Фримен А. Entity Framework Core 2 для ASP.NET Core MVC для профессионалов / Адам Фримен., 2019. – 624 с.

57. Введение в Entity Framework Core [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://metanit.com/sharp/entityframeworkcore/1.1.php>.

58. Флетчера П. Введение в SignalR [Электронный ресурс] / Патрик Флетчера // Microsoft. – 2014. – Режим доступа до ресурсу: <https://docs.microsoft.com/ru-ru/aspnet/signalr/overview/getting-started/introduction-to-signalr>.

59. Грофф Д. Р. SQL: полное руководство / Джеймс Р. Грофф., 2020. – 960 с.

60. SQL Server [Электронный ресурс] // Microsoft. – 2019. – Режим доступа до ресурсу: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2019>.

## **ДОДАТОК А Графічний матеріал**

## **ПЛАКАТ 1 Схе́ма структурна класі́в системи**

## **ПЛАКАТ 2 Схе́ма структурна пакеті́в системи**

### **ПЛАКАТ 3 Схема структурна бази даних**

**ПЛАКАТ 4 Графіки зміни множини рішень в процесі роботи  
алгоритму**

## **ПЛАКАТ 5 Екранні форми інформаційної системи**

**ПЛАКАТ 6 Схеми структурна варіантів використання**



**ПЛАКАТ 7 Блок-схема багатокритеріального генетичного  
алгоритму**